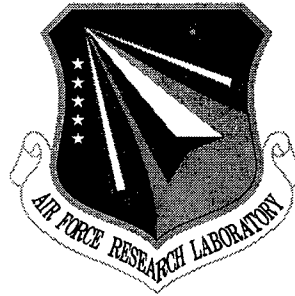


AFRL-IF-RS-TR-2001-114
Final Technical Report
June 2001



ASYNCHRONOUS TRANSFER MODE (ATM) USER SECURITY SERVICES

Odyssey Research Associates

Douglas Long and Peter Samsel

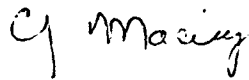
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

20010810 022

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2001-114 has been reviewed and is approved for publication.



APPROVED: CHESTER J. MACIAG
Project Engineer



FOR THE DIRECTOR: WARREN H. DEBANY, JR., Technical Advisor
Information Grid Division
Information Directorate

If your address has changed or if you wish to be removed from the Air Force Research Laboratory Rome Research Site mailing list, or if the addressee is no longer employed by your organization, please notify AFRL/IFGB, 525 Brooks Rd, Rome, NY 13441-4505. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE Jun 01		3. REPORT TYPE AND DATES COVERED Final Aug 94 - Jun 98
4. TITLE AND SUBTITLE ASYNCHRONOUS TRANSFER MODE (ATM) USER SECURITY SERVICES			5. FUNDING NUMBERS C - F30602-95-C-0234 PE - 62702F PR - 4519 TA - 22 WU - 41	
6. AUTHOR(S) Douglas Long and Peter Samsel				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Odyssey Research Associates Cornell Business & Technology Park 33 Thornwood Dr, Suite 500 Ithaca, NY 14850-1250			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFRL/IFGB 525 Brooks Road Rome, NY 13441-4505			10. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2001-114	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Chester Maciag, IFGB, 315-330-3184				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) ATM USA provides the framework for building security solutions for DOD ATM users. This architecture can be configured to support encryption devices at the local workstation, at the ATM switch, and at enclave boundaries (including cell and link encryptors). The enclave security policy can be used to provide a fine grain control over who has access to ATM services, the type and nature of the services that can be accessed, when they can be accessed, etc. The policy can also provide flexibility to balance the quality of services with the quality of protection for a connection and can provide dynamic management of both quality of service and quality of protection. The ATM USA is also compatible with emerging ATM standards, providing a solid basis for future compatibility as these standards develop in the future. In addition, we have specified the ATM USA security extension to the ATM Native Services APL. In addition to specifying the two API primitives, we defined how these primitives are invoked by an application in conjunction with its Connection Manager to manage the security services.				
14. SUBJECT TERMS ATM User Security Services			15. NUMBER OF PAGES 120	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Table of Contents

1	Introduction	1
1.1	Quality of Protection Management	1
1.2	Quality of Service Management	1
2	Requirements	4
2.1	ATM USS Statement of Work	4
2.1.1	ATM USS SOW: Section 4.1.2.1	4
2.1.2	ATM USS SOW: Section 4.1.2.2	4
2.1.3	ATM USS SOW: Section 4.1.2.3	4
2.2	Derived Requirements	5
3	Architecture	8
3.1	Heterogeneous ATM Networks	8
3.2	ATM User Security Architecture Overview	9
3.2.1	Security Service Application Levels	11
3.2.1.1	End-to-End Security by the Application:	13
3.2.1.2	End-to-End Security by the Connection Manager:	14
3.2.1.3	End-to-End Security by the End Switch:	14
3.2.1.4	Enclave-to-Enclave Security by the Gateway:	15
3.2.1.5	Link-to-Link Security by the Gateway:	16
3.3	User Workstations	17
3.3.1	Applications	18
3.3.2	Connection Manager	18
3.3.2.1	Connection Manager Control Plane	19
3.3.2.1.1	Application Control	19
3.3.2.1.2	Security Manager Interface	20
3.3.2.1.3	Path Control	21
3.3.2.2	Connection Manager User Plane	21
3.3.2.2.1	Application Control	21
3.3.2.2.2	Security Manager Interface	21
3.3.2.2.3	Path Control	22
3.3.3	Local Security Services	22
3.4	Security Management Workstation	22
3.4.1	Security Manager	23
3.4.1.1	Security Manager Control Plane	24
3.4.1.1.1	Connection Manager Interface	24
3.4.1.1.2	Path Selection Control	24
3.4.1.1.3	Network Security Manager	24
3.4.1.1.4	Security Policy Control	25
3.4.1.1.5	Security Policy Manager	25
3.4.1.1.6	Decision Control	25
3.4.1.2	Security Manager Management Plane	26
3.4.2	Security Policy Manager	28
3.4.3	Network Security Manager	28
3.4.4	QoS Path Manager	28
3.4.5	Directory Services	30

3.4.6	Local Security Services.....	30
3.5	Gateways.....	30
3.6	Network Management.....	31
4	Interfaces.....	32
4.1	User Security Interface.....	32
4.1.1	ATM Native Security Services.....	33
4.1.1.1	The ATM Forum Specification.....	33
4.1.1.2	ATM USA Security Extensions.....	36
4.1.2	Windows Sockets.....	38
4.1.2.1	The Windows Socket 2 API Specification.....	38
4.1.2.2	ATM USA Security Extensions.....	39
4.1.2.2.1	CMSetsockopt().....	40
4.1.2.2.2	CMGetsockopt().....	42
4.1.2.3	Windows and the ATM USA.....	42
4.1.3	ATM USA Security Parameters.....	44
4.1.3.1	Identification.....	45
4.1.3.2	Security Services.....	45
4.1.3.2.1	Confidentiality.....	46
4.1.3.2.2	Integrity.....	47
4.1.3.2.3	Authentication.....	48
4.1.3.2.4	KeyExchange.....	49
4.1.3.2.5	Certificate Exchange.....	50
4.1.3.2.6	Key Update.....	51
4.1.3.2.7	Access Control.....	52
4.1.3.3	Routing.....	53
4.2	Client Workstation to Security Management Workstation Interface.....	53
4.2.1	SM_outgoing_call_request.....	54
4.2.2	SM_outgoing_call_confirm.....	54
4.2.3	SM_wait_on_incoming_call_request.....	54
4.2.4	SM_arrival_incoming_call_request.....	55
4.2.5	SM_arrival_incoming_call_confirm.....	55
4.2.6	SM_accept_incoming_call_request.....	55
4.2.7	SM_accept_incoming_call_confirm.....	55
4.2.8	SM_call_active.....	55
4.2.9	SM_call_release.....	56
4.2.10	SM_add_party_request.....	56
4.2.11	SM_add_party_confirm.....	56
4.2.12	SM_add_party_success.....	56
4.2.13	SM_drop_party_request.....	56
4.2.14	SM_drop_party_confirm.....	57
4.2.15	SM_drop_party_success.....	57
4.3	Security and Signaling Coordination.....	57
4.3.1	Point-to-Point and Point-to-Multipoint Connection Setup.....	59
4.3.2	Point-to-Point and Point-to-Multipoint Connection Acceptance.....	61
4.3.3	Adding a Party to a Point-to-Multipoint Connection at the Root Node.....	64
4.3.4	Leaf Initiated Join at the Root Node.....	66

4.3.5	Leaf Initiated Join at a Leaf Node.....	67
4.3.6	Sending and Receiving Data	69
4.3.7	Application Initiated Connection Release.....	70
4.3.8	Network Initiated Connection Release.....	70
4.3.9	Dropping a Party from a Point-to-Multipoint Connection.....	71
4.4	Security Management Workstation to Enclave Boundary Interface.....	73
4.4.1	GW_open_control	73
4.4.2	GW_open_data.....	73
4.4.3	GW_close_control.....	73
4.4.4	GW_close_data	73
4.5	Security and Signaling Coordination with Gateway.....	74
4.5.1	Setup of Outgoing Connection.....	74
4.5.2	Setup of Incoming Connection.....	75
4.5.3	Leaf Initiated Join at a Leaf Node.....	77
4.5.4	Outgoing Release of Established Connection.....	79
4.5.5	Incoming Release of Established Connection.....	80
4.6	Security Management Workstation to Security Management Workstation Interface.....	81
4.7	Security Manager to FASTLANE Gateway Device Interface.....	81
4.8	An Overview of ATM PNNI Standard	82
5	Conclusion.....	86
6	Glossary.....	87
7	References	90
8	Appendix - ATM Requirements Collection.....	93
8.1	Defense Information System Network Security Architecture (DISN).....	93
8.2	Multilevel Information System Security Initiative (MISSI).....	96
8.3	Theater Battle Management (TBM).....	98
8.4	Global Grid Security Architecture	103
8.5	Derived Requirements.....	107

List of Figures

Figure 1. Heterogeneous ATM Networks.....	9
Figure 2. ATM USA Components within an Enclave	10
Figure 3. Secure Communications between Security Agents	11
Figure 4. End-to-End Security by the Application.....	13
Figure 5. End-to-End Security by the Connection Manager.....	14
Figure 6. End-to-End Security by the End Switch.....	15
Figure 7. Enclave-to-Enclave Security by the Gateway.....	16
Figure 8. Link-to-Link Security by the Gateway.....	17
Figure 9. ATM USA Connection Manager Architecture.....	18
Figure 10. Security Manager Control Plane.....	23
Figure 11. Security Manager Management Plane.....	27
Figure 12. Native ATM Services Reference Model.....	34
Figure 13. ATM Native Services State Diagram	35
Figure 14. Setting Connection Attributes.....	36
Figure 15. Security Reference Model	37
Figure 16. Setting Security Attributes.....	37
Figure 17. WinSock2 Generic Architecture.....	43
Figure 18. Network Security Shim Architecture.....	58
Figure 19. Making a Point-to-Point Call.....	60
Figure 20. Security Failure for Point-to-Point Call.....	61
Figure 21. Accepting a Point-to-Point Call.....	62
Figure 22. Security Rejection of Point-to-Point Call.....	64
Figure 23. Adding a Party to Point-to-Multipoint Connection.....	65
Figure 24. Security Rejection of Adding a Party.....	66
Figure 25. Leaf Initiated Join at the Root Node.....	67
Figure 26. Security Rejection of a Leaf Initiated Join	67
Figure 27. Leaf Initiated Join at a Leaf Node	68
Figure 28. Security Rejection of Leaf Initiated Join at a Leaf Node	69
Figure 29. Sending and receiving data on a connection.....	70
Figure 30. Connection Release by an Application.....	71

Figure 31. Connection Release from the network.....	71
Figure 32. Dropping a Party from a Multipoint Connection.....	72
Figure 33. Outgoing Connection Setup.....	75
Figure 34. Outgoing Connection Setup Rejection	75
Figure 35. Incoming Connection Setup.....	77
Figure 36. Incoming Connection Setup Rejection.....	77
Figure 37. Leaf Initiated Join at a Leaf Node	78
Figure 38. Leaf Initiated Join at a Leaf Node Rejection.....	79
Figure 39. Outgoing Connection Release	80
Figure 40. Incoming Connection Release.....	81
Figure 41. PNNI Network Hierarchical Abstraction.....	84

List of Tables

Table 1. Windows Sockets 2 Mappings.....	39
Table 2. CMSetsockopt and CMGetsockopt Parameters.....	41
Table 3. Identification Parameters	45
Table 4. Confidentiality Parameters.....	46
Table 5. Integrity Parameters	47
Table 6. Authentication Parameters	48
Table 7. Key Exchange Parameters.....	49
Table 8. Certificate Exchange Parameters	50
Table 9. Key Update Parameters.....	51
Table 10. Access Control Parameters	52
Table 11. Routing Parameters	53
Table 12. Connection Manager to Security Manager Interface Primitives.....	54
Table 13. Security Manager to Gateway Device Interface Primitives.....	73
Table 14. DISN Requirements	93
Table 15. MISSI Requirements.....	96
Table 16. Theater Battle Management Requirements.....	98
Table 17. Global Grid Security Architecture Requirements.....	103
Table 18. Derived Requirements Traceability Matrix	107

1 Introduction

This document describes the Asynchronous Transfer Mode (ATM) User Security Architecture developed by ORA for the US Air Force Rome Lab under Contract No. F30602-95-C-0234. This architecture is designed to meet the user requirements for security services for DoD ATM networks. The requirements for this architecture were derived from the security requirements of four programs designated by the Air Force: the Global Grid Security Architecture, the Defense Information System Network Security Architecture (DISN), the Multilevel Information System Security Initiative (MISSI), and the Theater Battle Management C4I Architecture for Deployable Operations. These derived requirements are described in Section 2.

Terms specific to this document are defined in the Glossary.

1.1 Quality of Protection Management

The fundamental aim of the ATM User Security Architecture (USA) is the management of security services for ATM networks. Such basic security services include privacy, integrity, authentication, non-repudiation, access control, and auditing. In addition to managing basic security services, the architecture supports gateway device management, network management and dynamic routing.

ATM USA has a flexible architecture that can provide the quality of protection required by the DoD while at the same time maintaining compatibility with commercial industry security standards. To achieve this flexibility, ATM USA supports the application of security mechanisms at different layers in the ATM protocol stack. Exactly what is allowed in a particular instantiation of ATM USA is governed by a security policy that is enforced by a security server that controls the establishment and operation of ATM connections.

1.2 Quality of Service Management

ATM technology offers the opportunity to construct networks that can carry many types of services. This may be achieved through the flexibility ATM offers in supporting diverse bandwidths and qualities of service (QoS). This flexibility, however, introduces the need for more complex management and control than that required for a conventional single service network. In addition, the status of network resources is constantly changing – due to applications that can change QoS requirements during application execution (e.g., an interactive multimedia application), loss of system resources, burst traffic that results in network congestion, or other causes – requiring QoS management schemes to be dynamic and adaptive to these changes.

Most of the current research on ATM network QoS management has focused on degrading the QoS performance as the network conditions deteriorate. Very little has

been done to upgrade the QoS when the network load is light and performance can be improved. Furthermore, a new class of large applications with a wide range of QoS requirements is emerging. A general network management architecture must be sufficiently flexible to support positive and negative changes to QoS with changing network conditions and emerging network applications.

Many researchers have proposed new QoS extensions to address current limitations:

- An Open Systems Interconnection (OSI) QoS extension has been proposed to define new QoS terms, such as threshold, compulsory, and QoS interpretation. This extension adds delay variance to the QoS parameters and negotiates the QoS for existing applications without releasing the connection [1].
- A coherent relation between an application and the protocol layer, allowing the mapping of QoS between them, is essential in QoS management. A QoS framework has been proposed in which the QoS manager interacts with both the applications and the network management system. It achieved a coherent relation between the application and the protocol layer by having QoS-related functionality in each layer [2]. Similar approaches offer an integral framework for QoS specifications and resource control over all communication layers [3].
- Given that the status of the resources between source and destination hosts must be known in order to maintain the QoS for an application, it is desirable to schedule resources at the host level [4]. Using the network information provided by the network management system, a set of reliable, single direction, and out-of-band high-level QoS negotiation protocols has been proposed to dynamically manage the application QoS before and after application execution [5].
- A set of QoS negotiation procedures for distributed multimedia applications has also been described to optimize the application performance based on the cost constraints. It can support dynamic adaptation in reaction to QoS degradations without user intervention [6].
- Configurable domain-based Virtual Path Connections (VPCs) in ATM networks have been proposed as a means to reduce the size of both the virtual path routing table and the complexity of virtual path administration. Such an approach dynamically reconfigures the network to meet the QoS of particular applications [7]. In this virtual path approach, the main criterion is to ensure all virtual path connections satisfy the QoS requirements of the application.
- A virtual path routing scheme has been described, allowing the determination of a virtual path configuration and bandwidth assignment matching a given network condition, in order to minimize the loss of data during network failure [8].
- An ATM transport layer has been proposed to provide per-virtual circuit QoS. It can transfer reliable and unreliable data with a choice of feedback and leaky-bucket flow control using inexpensive personal computers [9].
- RSVP, a reservation protocol that provides QoS for multimedia information transmission over internet networks, has been proposed. RSVP is based on a

connection-oriented principle that resources are reserved in advance for a channel set-up. Data transmission then follows the same channel [10].

Dynamic management of VPC routes and resource allocations can be accomplished by continuously monitoring the network and reacting to repeated congestion patterns and topological changes caused by failures or the addition of network elements such as links and nodes. The standardization of the Simple Network Management Protocol (SNMP) and the ATM Management Information Base (MIB) provides one method to achieve this. However, the read/write operations of MIB variables are too slow to be used in high-speed networks. An alternative approach to gathering network state information by reading MIB variables is provided by the ATM Forum's PNNI routing protocol, which enables information about the network to be gathered and disseminated in a scalable manner [11]. Although the PNNI standard provides network-monitoring functionality, there is no solution for dynamic VPC management, which must consider the application's QoS, QoP and Fault-Tolerance requirements.

Most of the current research techniques have assumed the existence of internal network and communication protocol support. In these techniques, there are high-level application QoS issues that have not been addressed. Furthermore, the current research and implementations that address the dynamic management of QoS and QoP are rigid and not flexible enough to support emerging large-scale networks. In contrast, we specify in this report an approach to achieving dynamic management of QoS and QoP for ATM networks that is general and based on an open architecture, while also applicable to the real-time management of a wide range of network applications.

2 Requirements

As mentioned in the Introduction, we derived the requirements for the ATM USA from four separate programs designated by the Air Force. The extracted ATM User Security Services (USS) program requirements, included in the Statement of Work, are given in Section 2.1. The derived requirements and their descriptions are given in Section 2.2.

2.1 ATM USS Statement of Work

The following sections identify the applicable requirements extracted from the ATM USS Statement of Work (SOW). The section numbers of the original document are specified in each section title.

2.1.1 ATM USS SOW: Section 4.1.2.1

Section 4.1.2.1 of the ATM USS SOW specifies the following goals for this project:

Implement dynamic policy-based routing to mitigate security risks associated with transmission over particular links of paths in the network. The characterization of security risks shall include link hardware type, implementation of link encryption devices, and stages of national crisis/urgency. Network topology polling shall be used to determine the characterization of network connectivity. The ATM USA shall provide automatic set-up of connections supporting user's security requirements.

2.1.2 ATM USS SOW: Section 4.1.2.2

Section 4.1.2.2 of the ATM USS SOW specifies the following goals for this project:

Provide protocol or API to facilitate requests for security services from the end-user to the network security management entity. User security service requests may include data sensitivity, information about encryption already employed by the user application, identification and authorization responses, call requests, and data priority level. Network security responses shall include call setup information, call acknowledgment/denial of service messages, approved PVCs, SVCs, and/or VPIs, VCIs, and challenges for identification and authentication.

2.1.3 ATM USS SOW: Section 4.1.2.3

Section 4.1.2.3 of the ATM USS SOW specifies the following goals for this project:

Communicate with and control link encryption devices (i.e., KG-189) and cell encryption devices (i.e., FASTLANE, TACLANE).

2.2 Derived Requirements

In this section, we describe the requirements for the ATM USA derived from the ATM USS SOW requirements and the security requirements of the four Air Force designated programs specified in Section 1: the Global Grid Security Architecture, the Defense Information System Network Security Architecture (DISN), the Multilevel Information System Security Initiative (MISSI), and the Theater Battle Management C4I Architecture for Deployable Operations (TBM). A traceability matrix relating each derived requirement to the corresponding Air Force designated program requirements may be found in Section 8.5 in the Appendix. Below, we list each requirement number and the associated requirement description.

- ATM-1. General:** Communications security shall include Cryptographic Security, Transmission Security, and Emission Security. Policies for handling differing types of transmission mediums, and cryptographic devices will be based upon existing local policies, existing military standards, and all applicable laws and regulations. Compromising emanations shall be controlled within appropriate TEMPEST guidelines. Support for a wide variety of transmission technologies, algorithm types and cryptographic techniques will be accommodated. Whenever possible, end-to-end security shall be employed rather than "black box" solutions (especially in messaging applications). To accomplish the goals of reliability and reusability of tools and techniques, modular approaches shall be employed. The ATM USA shall be capable of supporting assurance arguments demonstrating that systems meet their functional security requirements and are capable of being accredited to a level of security commensurate with the classification of the information being processed.
- ATM-2. Privacy/Integrity:** All communications networks shall provide message privacy and integrity services in order to protect messages against undetected loss, data repetition, data insertion and unauthorized modification.
- ATM-3. Authentication/Non-Repudiation:** All communications networks shall provide user authentication and non-repudiation services, including the positive identification of all originators and recipients, along with their roles, responsibilities, and positive acknowledgment, and a consistent method for protection against user deniability.
- ATM-4. Access Control:** To protect against unauthorized access, systems shall implement access control places and mechanisms, providing positive identification of each individual or system for both local and remote access.
- ATM-5. Auditing:** The ATM USA shall support the auditing of security relevant events, as well as the collection and analysis of distributed audit

information, thereby providing detection and analysis of security compromises or threats. Audit information shall contain, at minimum, the identity of the involved systems, the date and time, and the type of event. Events to be audited shall include changes in network connectivity, attachments to and from the network, changes in the security parameters of system components, transmission in the clear of data that was expected to be encrypted (due to loss of synchronization), and violations of integrity requirements.

- ATM-6. Availability:** The ATM USA shall mitigate denial of service attacks and/or support high availability of services.
- ATM-7. Multilevel Security:** The ATM USA shall support the generation, exchange, and protection of information at multiple classification levels. Security labels shall be embedded in multiple layers of the ATM protocol (e.g., IP & ATM). Labels will be used for key determination, as well as upgrading and downgrading services. The ATM USA shall support maintenance of the set of classifications, categories, and other markings that each application should maintain. Examples of other markings include declassification conditions. At a minimum, the classification, category and markings will be identified. The ATM USA shall also maintain a subject's clearances, and enforce access control based on sensitivity labels and clearances.
- ATM-8. Firewalls/Guards:** Security guards and firewalls shall be employed at system perimeters in order to mitigate the risks of connections to affiliated systems. As such, they shall manage the security of upgrading and downgrading operations, supporting local access policies with regard to passing messages between systems at different security levels.
- ATM-9. User Security Services:** The ATM USA shall provide protocol or API extensions to facilitate requests for security services from the end-user to the network security management entity. User security service requests may include call requests, data sensitivity queries, data priority level queries, identification and authorization responses, and application-applied encryption requests. Network security responses shall include call setup information, call acknowledgment of service or denial of service messages, approved PVCs, SVCs, and/or VPIs, VCIs, and challenges for identification and authentication.
- ATM-10. Dynamic Routing:** The ATM USA shall implement dynamic policy-based routing to mitigate security risks associated with transmission over particular links of paths in the network. The characterization of security risks shall include link hardware type, implementation of link encryption devices, and stages of national crisis/urgency. Network topology polling shall be used to determine the characterization of network connectivity.

The ATM USA shall provide automatic setup of connections supporting user's security requirements.

- ATM-11. Network Management:** The ATM USA shall support a common, secure, and distributed network management protocol. Security of network management systems and information, along with management of the security mechanisms that protect user traffic, shall be in accordance with MIL-HDBK-2054-1351. Security management functions shall be supported across public and private networks and shall accommodate hierarchies of decentralization. Security management functions will include management of cryptographic key material, authentication information, and access control (including access to the entire network or critical parts of the network.).
- ATM-12. Compatibility:** The ATM USA shall support a variety of independent or integrated voice, data, video, facsimile, and imagery applications. The ATM USA must consider usability, interoperability, backward compatibility, and existing infrastructure. The ATM USA must be consistent with the DoD Goal Security Architecture.
- ATM-13. Interoperability:** The ATM USA shall support interoperability, providing interfaces to systems used by our allies. Such interfaces must be able to support the dissemination of information that is releasable to a given ally, as well as be able to and protect information that is not releasable to that ally.
- ATM-14. Encryption Device Management:** The ATM USA shall provide services to communicate with and control high-speed link encryption devices (i.e., KG-189) and high-speed cell encryption devices (i.e., FASTLANE, TACLANE).
- ATM-15. End-to-End Encrypted Voice Service:** AAL Type 1 continuous bit rate (CBR) service shall be used for access to and from ATM networks and shall be employed to preserve BCI, and thereby maintain cryptographic synchronization between calling and called secure voice terminals.

3 Architecture

In this section, we discuss the ATM User Security Architecture that we have developed to meet the requirements listed in Section 2.2. The architecture itself, along with those requirements, is primarily motivated by the need to provide security in the context of heterogeneous ATM networks, as discussed in Section 3.1. In Section 3.2, we give an overview of the ATM USA and its constituent components, as well as discussing the possible levels at which security services may be applied in the architecture. Finally, in Sections 3.3 through 3.6, we describe the nature and functionality of the architecture components.

3.1 Heterogeneous ATM Networks

One of the principal requirements for DoD networks, now and in the future, is that they be increasingly based on commercial carriers and decreasingly based on government owned and operated networks. Thus, a combination of private and public ATM networks will play an important role in providing the global, high bandwidth communications necessary to meet DoD needs. Providing security solutions for these heterogeneous networks requires separation of the different parts of the network based upon differing levels of trust. The designers, implementers and administrators of these networks must protect the boundaries between regions of different trust levels in order to protect the integrity of each region.

The network configuration in Figure 1 illustrates the canonical solution to meeting the security requirements of heterogeneous networks. In this example, a public network connects two private networks. From the standpoint of the organization that owns the private networks, the organization can exercise greater control over the users and resources of the private network than it can exercise over the users and resources of the public network. This organization protects its resources from the less trustworthy public network by placing protection devices at the trust boundary. These *gateways* control the flow of information into and out of the private networks according to an organizational security policy.

The use of a gateway to protect a private network forms a secure enclave, which, ideally, provides a measure of protection against the presumably less trustworthy public network. For ATM USA, a single security policy governs each secure enclave and ensures a uniform application of the security policy throughout the enclave.

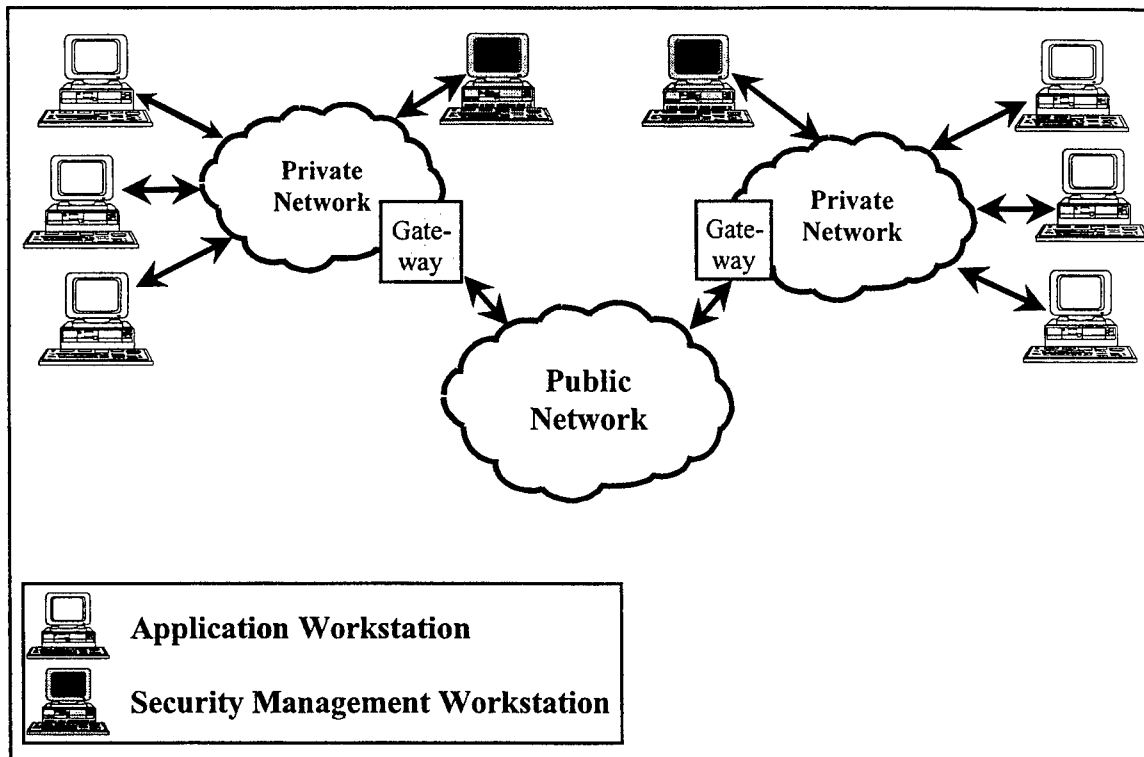


Figure 1. Heterogeneous ATM Networks

A gateway includes any device that filters data leaving the enclave (e.g., guards), filters data entering the enclave (e.g., firewalls), or encrypts outgoing data and decrypts incoming data (e.g., cell encryptors such as FASTLANE or TACLANE). Each type of device provides protection of the enclave in one way or another.

The laws, rules, and regulations of the organization that the enclave serves determine the security policy for a secure enclave. The Security Management Workstation enforces this security policy to the Security Manager, which in turn uses it to ensure that ATM connections are made and managed in accordance with the enclave security policy.

3.2 ATM User Security Architecture Overview

The ATM User Security Architecture (ATM USA) is designed to provide user security services for user applications that require ATM services. Figure 2 shows the major components of the ATM USA in a sample configuration within a secure enclave. Within this enclave, there may be user workstations (WS), each of which may be running one or more user applications. Also within the enclave is a Security Management Workstation (SMW), which defines and enforces the overall security policy for the enclave. Within the SMW, there are several major components: the Security Manager, the Security Policy Server, Directory Services, the Network Security Manager, and the QoS Path Manager. These figures also show other components (e.g., COTS or GOTS components) that interact with ATM USA components, but are not part of the ATM USA. These are Local Security Services, Network Management entities, and Gateways.

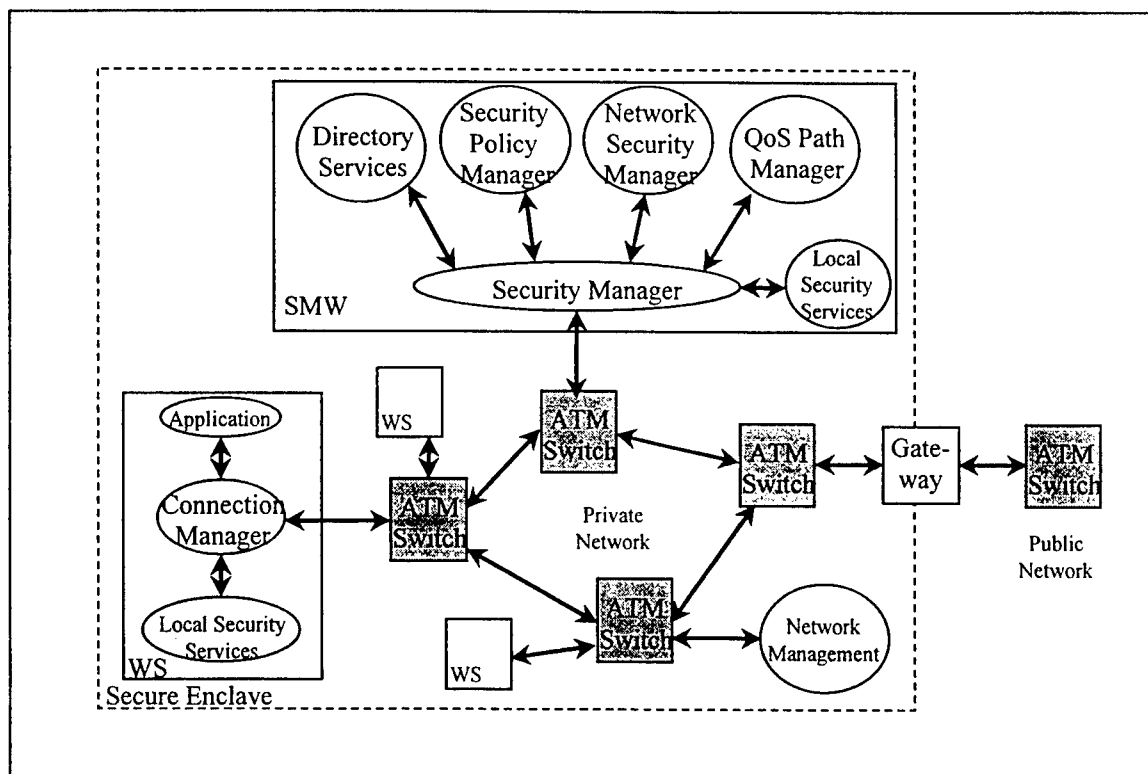


Figure 2. ATM USA Components within an Enclave

In this architecture, each ATM-enabled workstation will have a Connection Manager, which provides security services for all the ATM applications that run on the workstation. We assume that this workstation has a set of security services (supported by either hardware or software mechanisms) that can be used by applications running on the workstation. These may differ from workstation to workstation within the secure enclave. The Local Security Services component provides the interface to these security services.

Each enclave will have a Security Management Workstation (SMW) that will enforce the enclave security policy and govern the application of the security services necessary for policy enforcement. The key component of the SMW is the Security Manager, which is responsible for making security policy enforcement decisions, particularly as they relate to ATM connection establishment and management. The Security Manager is assisted by other components within the SMW: the Network Security Manager, QoS Path Manager, Network Management and Directory Services components. These components are primarily concerned with providing information to the Security Manager, especially information about the network outside the enclave.

ATM technology is a connection-oriented network technology. The ATM USA design provides security services to users based on the connection paradigm provided by ATM networks. There are two paramount issues in this architecture: First, is a network connection consistent with the overall philosophy of protection of the enclave? Failure to conform to the enclave policy may result in the loss of protected information or open the

enclave to potential penetration and attack. For this reason, each and every connection is examined and monitored to ensure that the connection should be allowed and to ensure it can securely opened and operated. Second, can the enclave properly support the security services that the connection requires? Proper support of a connection requires establishing the proper *security context* for the connection. A connection security context includes all information necessary to properly protect the data that the end users send over the connection. The end users of the connection must share the same security context to ensure that they can securely use the channel. The establishment of this shared security context may involve some negotiation and exchange of confidential information by the end users or security agents working on their behalf. The security context includes the choice of security services for the connection. These services may include end user authentication, confidentiality, integrity, data origin authentication, and so forth. The security context includes the choice of mechanisms or algorithms that are to be used to provide these services. In addition, the security context includes all necessary keys, certificates, and other information needed to support the chosen services and mechanisms.

3.2.1 Security Service Application Levels

The ATM USA is a flexible architecture that may allow the application of security services at different levels of the network architecture. The security policy enforced by the Security Management Workstation directly controls which level of security service application is allowed in the enclave, and under which circumstances it will be allowed.

At each level in the ATM USA, *security agents* provide security services. A security agent at one end of a secure connection is responsible for maintaining the appropriate security association with a peer security agent at the other end of the connection. A security agent is also responsible for application of security services to the data stream that is carried over the secure connection. For example, a security agent at one end of a connection may encrypt outgoing data being sent over the secure connection while the peer security agent at the other end of the connection decrypts the incoming data from the connection. This provides a secure data stream between the security agents, as shown by the solid line in Figure 3. This secure data stream can be transmitted through an untrusted communications channel, one that is subject to eavesdropping or other malicious intent.

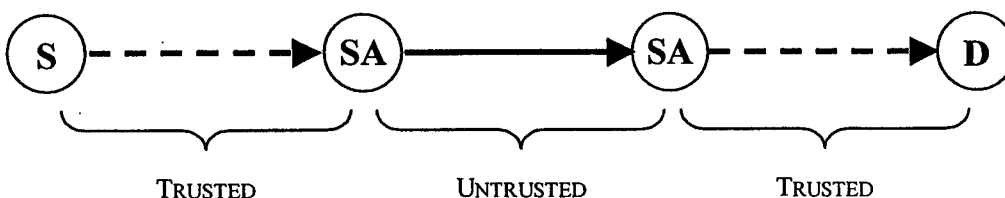


Figure 3. Secure Communications between Security Agents

The security agents can secure the data stream only while it is in transit between security agents. For the entire communications channel between the source and the destination to be secure, the data stream must also be protected between the source of the data (S) and the source's security agent, as well as between the destination's security agent and the destination (D). Typically, this protection is provided by physical collocation of the data source or data destination with the security agent for that source or destination within a trusted workstation or trusted enclave.

ATM USA provides for several different models for location of security agents:

End-to-End Security by the Application: The security agents are colocated with the applications, providing data security from the source to the destination. In this case, the applications must be trusted to properly use their security agents in the application of necessary security services to protect data being sent over the communications channel. The security policy may be used to limit which applications can be trusted, who may use them, the circumstances under which they may be used, etc. These applications may construct their own security context or may use the Connection Manager and Security Manager to construct the proper security context.

End-to-End Security by the Connection Manager: The security agents are colocated with the connection managers, providing data security from the source workstation to the destination workstation. Such security may be used when applications do not provide their own security support or cannot be trusted to do so in the context of the secure enclave. The connection managers on the respective workstations construct the security context for the applications and apply the security services to the connection on behalf of them. The application of the security services may be completely transparent to the application. The connection managers are then responsible for the protection of the data being sent over the communications channel.

End-to-End Security by the End Switch: The security agents are colocated with the ATM end switches connected to the source and destination workstations. In this case, the end switches are responsible for the protection of the data being sent over the communications channel. The connection managers and security managers ensure that applications establish or use connections with the proper security services. The security managers may also configure and manage the security services and security context.

Enclave-to-Enclave Security by the Gateway: The security agents are colocated with the gateways of the source and destination enclaves. In this case, the gateways are responsible for the protection of the data being sent over the communications channel. The connection managers and security managers ensure that applications establish or use connections that have the proper security services applied. The security managers may also configure and manage the security services and security context provided at the network level.

Link-to-Link Security by the Gateway: The security agents are colocated with the gateways of the source, destination and intermediate enclaves in the connection path. In

this case, the gateways at all of these enclaves are responsible for the protection of the data being sent over the communications channel. The connection managers and security managers ensure that applications establish or use connections with the proper security services. The security managers may also configure and manage the security services and security context provided at the network level.

We illustrate each of these five cases in the following sections.

3.2.1.1 End-to-End Security by the Application:

ATM USA supports End-to-End Security by the Application for the case where security agents (SA) reside at the application level at each end of a connection, as shown in Figure 4. These security agents act together to establish a security association for the desired security service. The application uses its security agent, with the appropriate security association, to provide the desired security service for the data stream passing over the connection.

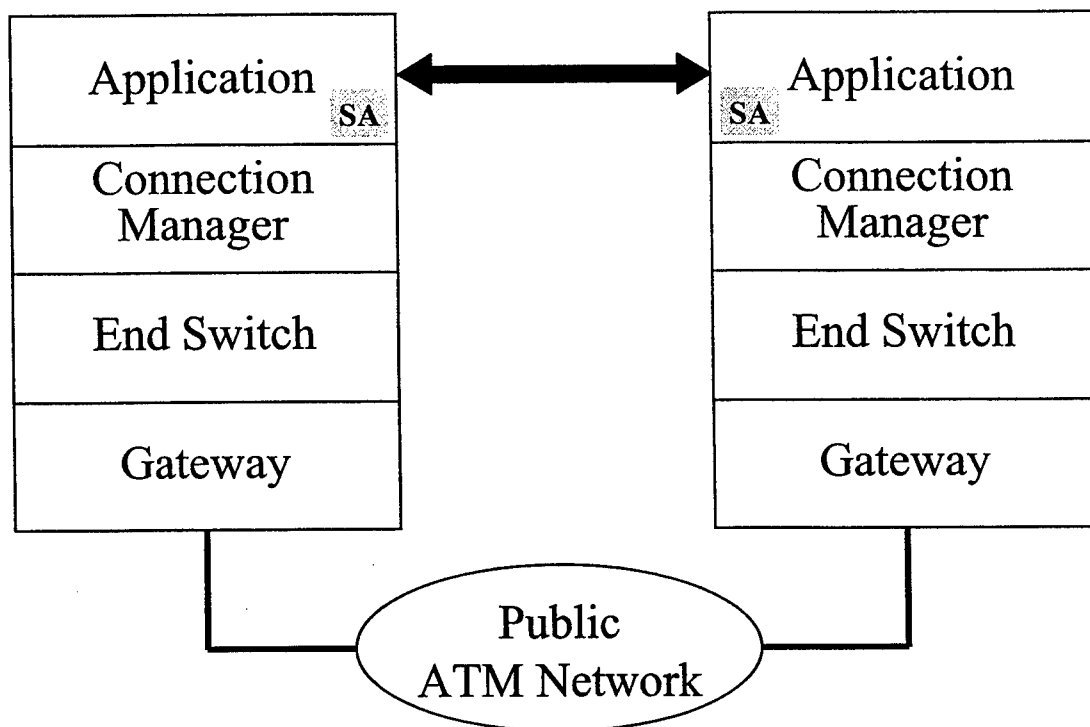


Figure 4. End-to-End Security by the Application

In this case, the application is responsible for providing the security service in question. The Connection Manager and the network do not provide any security services. The Connection Managers do however control the establishment of the connection between the applications and will not allow the connection if it does not meet the security policies of their respective enclaves.

3.2.1.2 End-to-End Security by the Connection Manager:

ATM USA supports End-to-End Security by the Connection Manager for the case where security agents (SA) reside at the Connection Manager level at each end of a connection. See Figure 5. These security agents act together to establish a security association for the desired security service. The Connection Manager uses its security agent, with the appropriate security association, to provide the desired security service for the data stream passing over the connection.

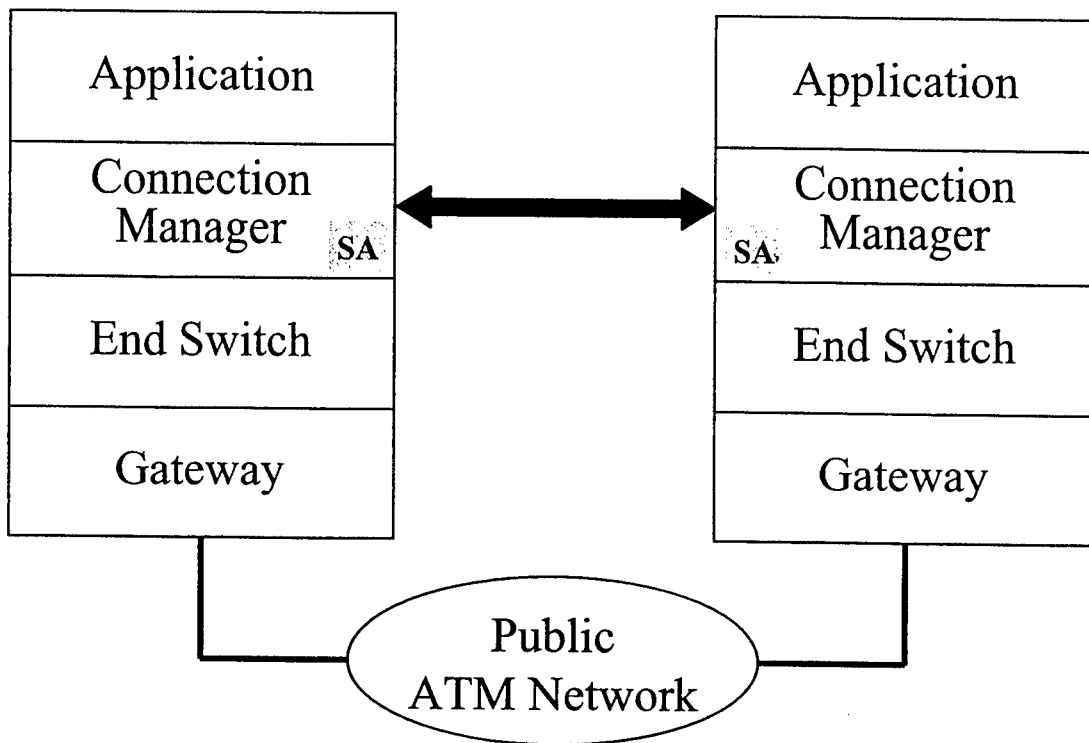


Figure 5. End-to-End Security by the Connection Manager

In this case, the Connection Manager is responsible for providing the security service in question. The application and the network do not provide any security services. The Connection Managers also control the establishment of the connection between the applications and will not allow the connection if it does not meet the security policies of both enclaves.

3.2.1.3 End-to-End Security by the End Switch:

ATM USA supports End-to-End Security by the End Switch for the case where security agents (SA) reside at the End Switch at each end of a connection. See Figure 6. These security agents act together to establish a security association for the desired security service. The End Switch uses its security agent, with the appropriate security association, to provide the desired security service for the data stream passing over the connection.

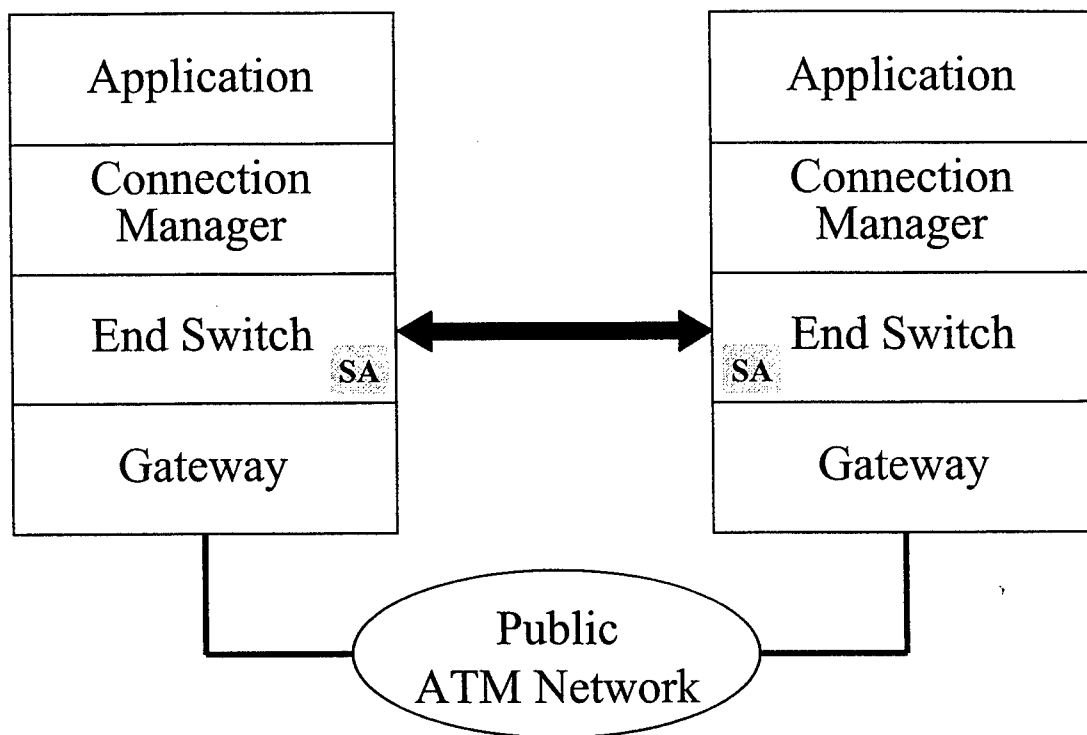


Figure 6. End-to-End Security by the End Switch

In this case, the End Switch is responsible for providing the security service in question. The application and the Connection Manager do not provide any security services. The Connection Managers also control the establishment of the connection between the applications and will not allow the connection if it does not meet the security policies of both enclaves.

3.2.1.4 Enclave-to-Enclave Security by the Gateway:

ATM USA supports End-to-End Security by the Gateway for the case where security agents (SA) reside at the enclave gateway at each end of a connection. See Figure 7. These security agents act together to establish a security association for the desired security service. The Gateway uses its security agent, with the appropriate security association, to provide the desired security service for the data stream passing over the connection.

In this case, the Gateway is responsible for providing the security service in question. The application and the Connection Manager do not provide any security services. The Connection Managers also control the establishment of the connection between the applications and will not allow the connection if it does not meet the security policies of both enclaves.

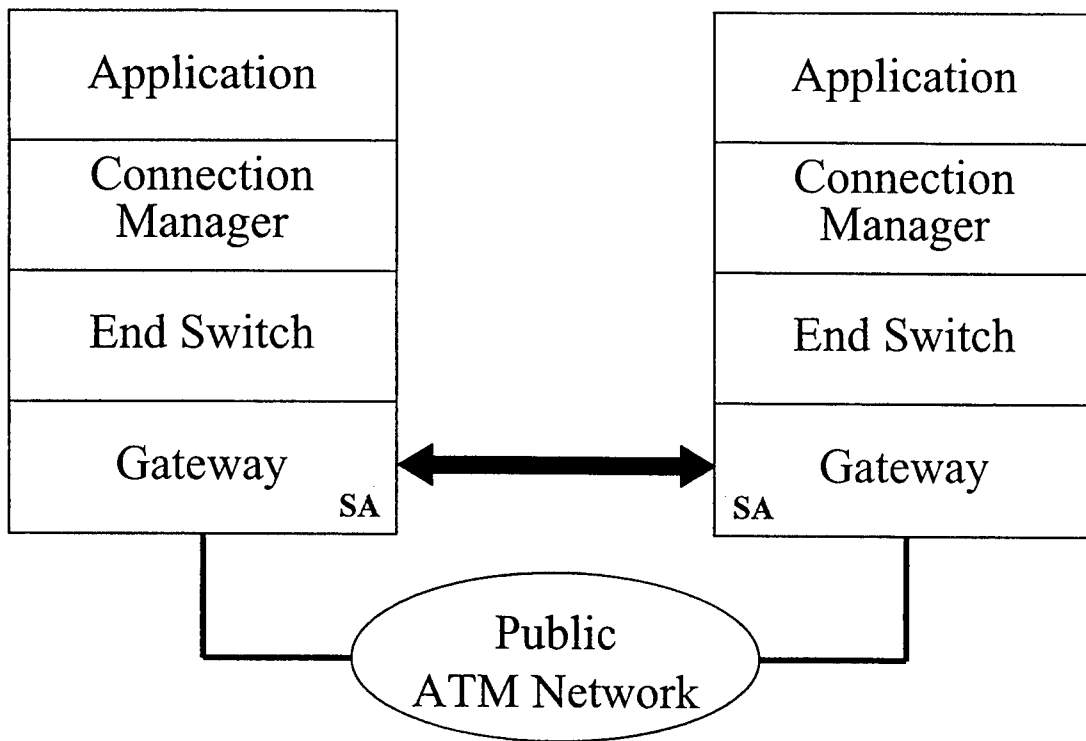


Figure 7. Enclave-to-Enclave Security by the Gateway

3.2.1.5 Link-to-Link Security by the Gateway:

ATM USA supports Link-to-Link Security by the Gateway for the case where security agents (SA) reside at the enclave gateway at each end of a connection and at intermediate enclaves. See Figure 8. A gateway uses its security agent, with the appropriate security association, to provide the desired security service for the data stream passing over the connection between trusted enclaves. The data on a connection may pass through intermediate trusted enclaves, but security services are not provided on the portion of the connection that passes through the trusted enclave.

In this case, the Gateway is responsible for providing the security service in question. The application and the Connection Manager do not provide any security services. The Connection Managers also control the establishment of the connection between the applications and will not allow the connection if it does not meet the security policies of both enclaves.

As discussed in Section 3.2 and shown in Figure 2, a secure enclave supported by the ATM USA is composed of several components: a number of user workstations, a security management workstation, a network management device, a gateway device, and various ATM switches. Of these, the user workstations and the security management workstation are in turn composed of several subcomponents, which are recognized by the architecture. In Sections 3.3-3.6, we discuss in detail the nature and functionality of components constituting the ATM USA, along with their various subcomponents.

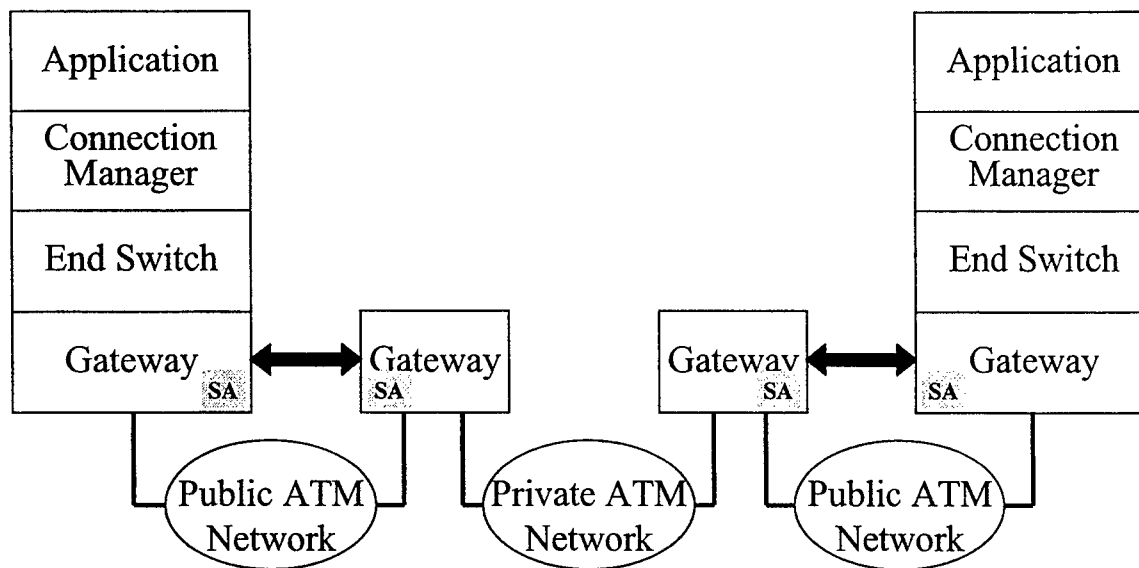


Figure 8. Link-to-Link Security by the Gateway

3.3 User Workstations

Each ATM-enabled user workstation has user Applications, Local Security Services, and a Connection Manager. The Connection Manager controls and coordinates the establishment and management of ATM connections, the transfer of data to and from ATM connections, and the application of appropriate security services to these connections for the user. Figure 9 shows the architecture of these ATM USA components.

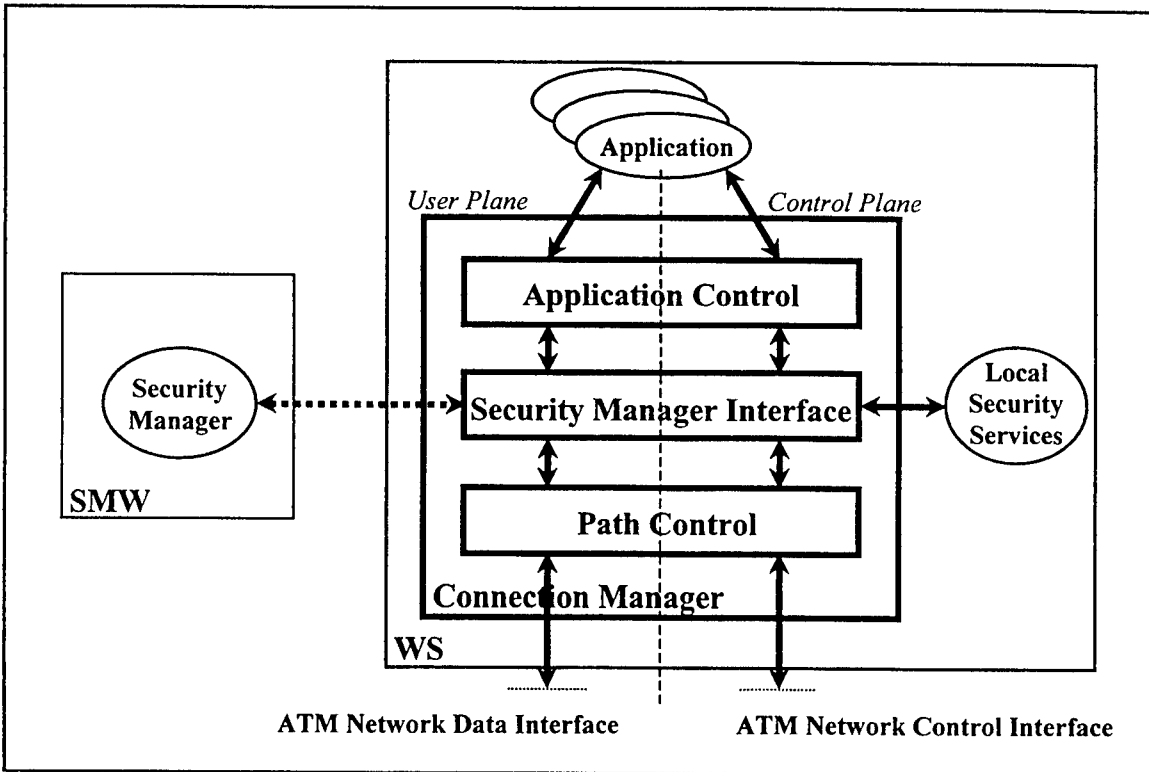


Figure 9. ATM USA Connection Manager Architecture

3.3.1 Applications

The Application component represents ATM user applications. Applications may be security unaware or security aware. Security unaware applications do not need to provide any quality of protection parameters; the enclave security policy provides all relevant security information. Security aware applications may provide quality of protection parameters, which are used to set up and manage the application's ATM connections. This allows security aware applications to take a more active role in their security. These applications are still governed, however, by the enclave security policy, and the quality of protection parameters they provide must be consistent with this policy.

3.3.2 Connection Manager

The Connection Manager resides between user applications and the ATM network interface, and controls the establishment, security management, and termination of all ATM connections for the applications according to the enclave security policy. The Connection Manager consists of two separate parts, a User Plane and a Control Plane. The User Plane is responsible for the transmission of user data between the user application and the network interface, and the application of any security services that Security Manager or user application requires the Connection Manager to apply. The Control Plane is responsible for connection establishment, management, and termination.

Note that the Connection Manager has three main components: the Application Control, the Security Manager Interface, and the Path Control. Each of these components is present in both the User and the Control Plane. The Application Control is responsible for communications with user applications. The Security Manager Interface is responsible for communication with the Security Manager and the application of any necessary security services. The Path Control is responsible for communication with the ATM network interface.

The User Plane and the Control Plane clearly separate functions for data transmission from functions for connection control. User applications can use the control functions of the Control Plane, which are under the direct control of the Security Manager, only to open or close an ATM connection. User applications cannot use the data transmission functions in the User Plane to cause the establishment or termination of a connection.

Sections 3.3.2.1 through 3.3.2.2 describe the User and Control Planes in more detail.

3.3.2.1 Connection Manager Control Plane

The Connection Manager Control Plane is responsible for connection establishment, management, and termination. The Security Management Workstation (SMW) supervises all the Connection Manager's control functions, which ensures that the Connection Manager acts in accordance with the enclave security policy. The Connection Manager communicates all connection requests to the SMW, which acts on them according to the enclave security policy and instructs the Connection Manager on how to proceed. The Connection Manager then attempts to establish the connection according to the instructions of the SMW. Furthermore, the SMW may instruct the Connection Manager to provide or negotiate a security context for a connection.

3.3.2.1.1 Application Control

The Application Control is the Connection Manager component that is responsible for interfacing with user applications. This component provides applications with a control interface for setting up and shutting down ATM connections. The application may specify user security requirements for a connection as part of connection establishment.

To set up an ATM connection, the application uses the Application Control's control interface to approach the Connection Manager with its requirements for the desired connection. These requirements will consist of the source and destination address, the quality of service (QoS) requirements, and the quality of protection (QoP) requirements necessary to set up a connection. The Application Control passes the request to the SMW Interface and waits for the response. If the connection request is successful, the Application Control receives connection information from the Security Manager interface and passes it to the application. Otherwise, the Application Control informs the application of the failure of the connection request.

To shut down an ATM connection, the application uses the Application Control's control interface to inform the Connection Manager of its desire to close the connection. The Application Control passes the request on to the Security Manager Interface and waits for a response. The Application Control passes the Security Manager Interface's response back to the Application.

3.3.2.1.2 Security Manager Interface

In the Control Plane, the Security Manager Interface is responsible for communication with the Security Manager. Note that, in the Control Plane, the Security Manager Interface may use local security services to properly secure communications with the Security Manager.

The Security Manager Interface will pass the connection request from the Application Control, together with all the connection parameters, to the Security Manager in the Security Management Workstation. The Security Manager will decide if the security policy allows the establishment of the connection. If the Security Manager denies the connection request, the Security Manager Interface passes the response to the Application Control. If the Security Manager allows the connection, then Security Manager Interface passes the connection request to the Path Control and waits for a response. The Path Control attempts to establish the connection and returns the response to the Security Manager Interface, which passes it to the Application Control. If the Path Control can set up the connection, this response will include connection information.

The Security Manager Interface may also be responsible for negotiating a security context for the connection. The SMI may use the in-band security message exchange and negotiation protocols defined in the ATM Security Specification, Version 1.0, or may use some other security exchange and negotiation protocols. Once the Path Control has established a connection, the Security Manager Interface uses the newly opened connection to establish the security context through a message exchange with its counterpart at the other end of the connection. After the Security Manager Interface has established the required security context, it makes the connection available for use by the application. The Security Manager Interface may use local security services, accessed through the Local Security Services component, to generate and process the security tokens. The enclave security policy governs the negotiation of security contexts by the Security Manager Interface.

The Security Manager Interface may also assist the Path Control to establish the security context using the signaling-based security message exchange and negotiation protocols defined in the ATM Security Specification, Version 1.0 [12]. In this case, the SMI generates and interprets the security tokens required by this exchange for the Path Control.

The Security Manager Interface will also monitor and manage the connection during its entire duration. For example, the Security Manager may instruct the Security Manager Interface to close a connection for security reasons. In this case, the Security Manager

Interface does not allow any further data to pass to or from the connection. The Security Manager Interface may also instruct the Path Control to establish a new path for a connection and start using this new path for data; this process can be completely transparent to the Application.

3.3.2.1.3 Path Control

In the Control Plane, the Path Control is responsible for communication with the ATM network control interface to set up and shut down ATM connections. The Path Control informs the Security Manager Interface of the success or failure of its attempt to set up or close a connection. Connection information for the user application is included in responses to successful requests to open a connection.

Optionally, the Path Control may be instructed to establish some, or all, of the security context, using the signaling-based security message exchange and negotiation protocols defined in the ATM Security Specification, Version 1.0 [12]. The Path Control accomplishes this message exchange as part of the connection setup. In this case, security tokens are included as part of the SETUP, CONNECT, CONNECT-ACKNOWLEDGE exchange. The Path Control uses security tokens provided by the Security Manager Interface for the SETUP and CONNECT-ACKNOWLEDGE messages and passes the token received with the CONNECT message to the Security Manager Interface.

3.3.2.2 Connection Manager User Plane

The Connection Manager User Plane is responsible for transmitting data between the application and the ATM network interface on established connections. Optionally, the Connection Manager User Plane may apply security services to data as it passes from application to network or from network to application on an established connection. Note that data must be passed through the Connection Manager with as low an overhead as possible (e.g., by passing pointers to blocks of data instead of copying blocks of data) to provide minimum data latency.

3.3.2.2.1 Application Control

In the User Plane, the Application Control provides applications with a data interface for sending data to and from established connections. The Application Control passes user data directly from the application to the Security Manager Interface and from the Security Manager Interface to the application.

3.3.2.2.2 Security Manager Interface

In the User Plane, the Security Manager Interface is primarily responsible for the application of any security services that the enclave security policy or application requires the Connection Manager to apply. For outbound data, the Security Manager Interface applies the security services to the data when it receives the data from the

Application Control and then passes the data to the Path Control. For the inbound data, the Security Manager Interface applies the security services to the data when it receives the data from the Path Control and then passes the data to the Application Control.

The Security Manager Interface will also immediately stop data transmission if the Security Manager instructs it to close an open connection.

3.3.2.2.3 Path Control

In the User Plane, the Path Control is responsible for passing data between the Security Manager Interface and the ATM network data interface.

3.3.3 Local Security Services

Each workstation has a set of security services (supported by either hardware or software mechanisms) that can be used either by applications running on the workstation or by the Connection Manager. These security services may differ from workstation to workstation within the secure enclave. The Local Services component provides the interface to these security services; the Security Manager Interface uses this interface to apply any needed security services.

The FORTEZZA card is an example of a hardware device that can provide security services to a user workstation.

3.4 Security Management Workstation

The Security Management Workstation consists of six principal components: the Security Manager, the QoS Path Manager, the Network Security Manager, the Directory Services, the Security Policy Manager, and the Local Security Services. The Security Manager is the principal component of the Security Management Workstation.

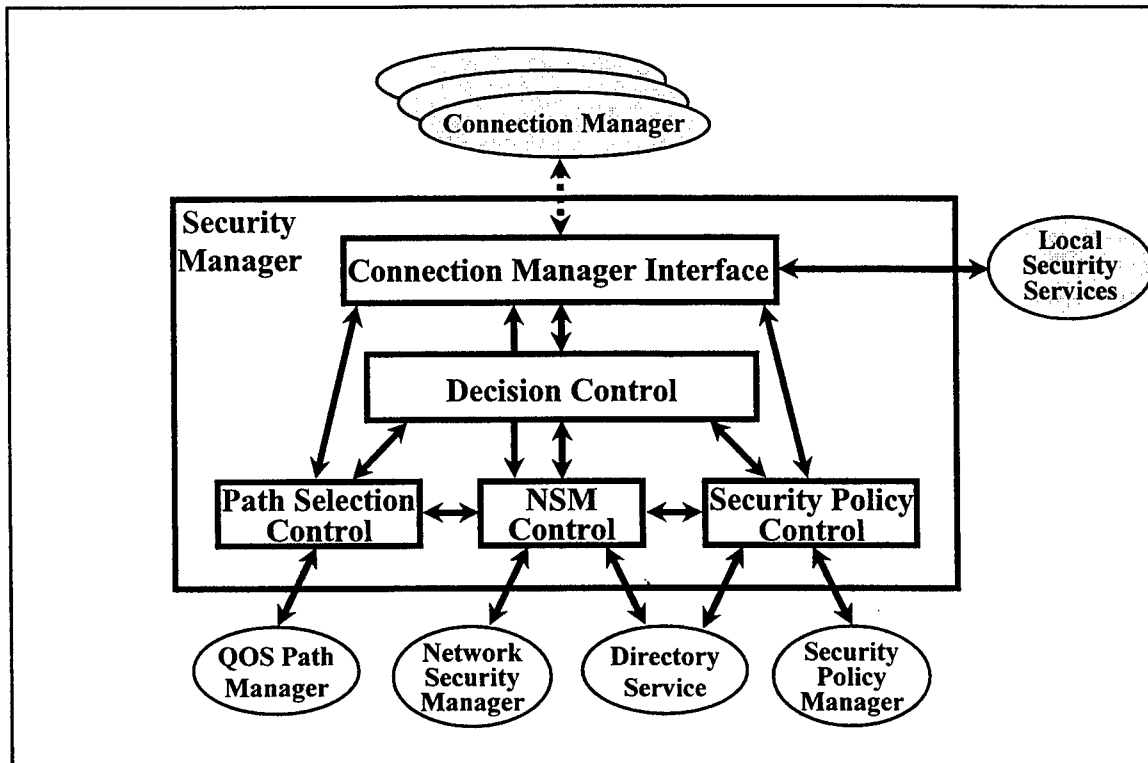


Figure 10. Security Manager Control Plane

3.4.1 Security Manager

The Security Manager is the ATM USA component responsible for security policy enforcement for all ATM connections to ATM equipped workstations within the enclave. The Security Manager consists of two separate parts, a Control Plane and a Management Plane. The Control Plane of the Security Manager controls the actions of the Connection Manager on each user workstation within the enclave. The Management Plane provides security management for the other components of the Security Management Workstation so that they can securely access resources outside of the enclave.

The design of the Security Manager for this architecture has been further broken down into six main components. These components are the Connection Manager Interface, Management Control, Path Selection Control, Network Security Manager Control, Security Policy Control, and Decision Control. Note that the Connection Manager Interface is present only in the Control Plane and the Management Control is present only in the Management Plane. Figure 10 shows the Security Manager Control Plane. Figure 11 shows the Security Manager Management Plane.

The Control Plane and the management plane clearly separate functions for connection control from functions for security management of the SMW. User applications can interact with the Control Plane only indirectly, as a side effect of requests to open or close a connection. User applications cannot use these control requests to affect the security management of the SMW itself.

Sections 3.4.1.1 through 3.4.1.2 describe the Control and Management Planes in more detail.

3.4.1.1 Security Manager Control Plane

The Security Manager Control Plane is responsible for interacting with the Connection Managers on user workstations to control user connections in accordance with the enclave security policy.

3.4.1.1.1 Connection Manager Interface

In the Security Manager Control Plane, the Connection Manager Interface is responsible for interfacing with the Connection Managers on all the user workstations in the enclave. The Connection Manager Interface receives requests from the Connection Managers and passes them onto the other components of the Security Manager. Once the Decision Control has decided on the disposition of a request, the Connection Manager Interface passes the response back to the Connection Manager that initiated the original request.

The communications between the Connection Managers and the Security Manager must be secure, requiring, at minimum, mutual authentication and integrity. The Connection Manager Interface applies the appropriate security services to these communications using the local security services provided on the SMW.

3.4.1.1.2 Path Selection Control

In the Control Plane, the Path Selection Control is the component of the Security Manager that is responsible for communication with the QoS Path Manager. When the Path Selection Control receives a request to set up a connection from the Connection Manager Interface, it passes the request on to the QoS Path Manager. The QoS Path Manager responds with a set of paths from the given source to the destination that satisfy the requested QoS. When it receives this list from the QoS Path Manager, the Path Selection Control passes it on to the Network Security Manager (NSM) Control and to the Decision Control.

3.4.1.1.3 Network Security Manager

In the Control Plane, the Network Security Manager (NSM) Control is responsible for communication with the Network Security Manager.

During connection establishment, the NSM Control will interact with the Network Security Manager to obtain all relevant security details about each of the paths provided by the Path Selection Control. This information will be simplified and forwarded to the Decision Control and to the Security Policy Control.

The NSM Control will consult with Directory Services to obtain information about various entities involved in the connection that is being set up.

The NSM Control will also interact with the Network Security Manager to provide security status updates on established connections. The Network Security Manager passes any status changes to the NSM Control, which forwards them to the Decision Control for action.

3.4.1.1.4 Security Policy Control

In the Control Plane, the Security Policy Control is responsible for downloading security policy information from the Security Policy Manager and for passing security policy information to the Decision Control.

During connection establishment, the Connection Manager Interface will pass information about a requested connection directly to the Security Policy Control. The NSM Control passes the information about the possible paths and their security characteristics to the Security Policy Control. The Security Policy Control queries the Security Policy Manager for security policy information relevant to the connection and then passes this security policy information about the proposed connection and paths to the Decision Control.

The Security Policy Control will also consult with Directory Services to obtain information about various entities involved in the connection that is being set up.

3.4.1.1.5 Security Policy Manager

During connection management, the enclave security policy stored in the Security Policy Manager also provides guidance about how to respond to certain changes in the security status of a connection.

3.4.1.1.6 Decision Control

In the Control Plane, the Decision Control is responsible for deciding if a connection request can be satisfied in a way that meets the enclave security policy, choosing a path for the connection from among the alternatives, directing the Connection Manager that made the original connection request to open the connection using the selected path, and providing security context information to the Connection Manager. The Decision Control receives inputs from the Connection Manager Interface, the Path Selection Control, the NSM Control, and the Security Policy Control. When the Decision Control has received all the necessary inputs from each of these components, it will reconcile all the information and make an intelligent choice regarding the paths, selecting a path that is consistent with the local security policy. If the Decision Control finds a satisfactory path, it informs and directs the Connection Manager to set up the connection. If the Decision Control does not find a satisfactory path, it directs the Connection Manager to refuse to set up the connection.

The Decision Control is also responsible for deciding how to respond to changes of the security status of current connections. The Network Security Manager passes status

changes to the NSM Control, which passes them to the Decision Control for action. Possible courses of action include the immediate termination of the connection, the acceptance of reduced quality of service in return for maintaining the desired quality of protection, and the establishment of an alternative path for the connection.

The basic function of the Decision Control is to make decisions based on inputs gathered by other ATM USA components. Since the decisions made by this component are critical to the correct functioning of the Security Manager and the correct enforcement of the enclave security policy, the functions of this component have been kept to a minimum. This will allow the designers and implementers of this component to subject it to careful analysis to ensure its correct functioning.

The Decision Control will use a rule-based decision mechanism. This rule-based decision mechanism will provide considerable flexibility during connection setup, including balancing quality of service considerations with quality of protection considerations. The enclave security policy controls all trade-off decisions; only trade-offs that are specified in the security policy are allowed. For example, many times the network QoS requirements and the QoP requirements for an application may be in conflict with the services the network may provide. In this case, the local security policy may allow the connection manager to make a trade-off between QoS requirements and QoP requirements. The connection manager may decide to compromise on one of the requirements, for example, accepting lower QoS in exchange for increased security. In addition, our approach allows the connection manager to preempt another connection based on priority levels. The security policy also controls this capability.

A generic rule-based expert system consists of a working memory, a rule-base, and a control procedure. The rule base contains all the rules about the problem domain. It is the knowledge base where the “expertise” of the system is stored. The knowledge base is represented as a set of actions. The expert system’s problem-solving capabilities are directly dependent on the number of rules it has in its rule-base. By increasing the number of these rules, the expert system can solve problems that are more complicated. The working memory contains the rules that are directly applicable to the problem under consideration. These are the rules that need to be “fired”, or executed, since it is these rules whose “if” clauses have been satisfied for the case at hand. In case of ambiguities, such as multiple rules that can be fired in the given scenario, the control procedure determines which rule will actually be fired. The system updates the working memory by asserting, modifying, or retracting working memory elements.

3.4.1.2 Security Manager Management Plane

The Security Manager Management Plane is responsible for providing and managing secure connections from the Security Management Workstation components that require connections to entities outside the enclave. The components that require this connectivity are the QoS Path Manager and the Network Security Manager. The Security Manager Management Plane is also responsible for providing secure connections to gateway devices that support remote management of their security policies. This capability allows

the Security Manager to coordinate the policies of the gateway devices with the overall enclave policy that the Security Manager enforces.

The QoS Path Manager and Network Security Manager may initiate connections as follows. The manager formulates the connection request and sends it to the corresponding management plane controller in the Security Manager. (I.e., the management plane Path Selection Control or the management plane NSM Control.) The receiving controller passes the connection request to the management plane Security Policy Control and to the management plane Decision Control. On receiving the connection request, the Security Policy Control passes the relevant security policy information to the Decision Control. The Decision Control either rejects the connection or passes it to the Management Path Control to open the appropriate connection. Either the Management Path Control succeeds in opening the connection, in which case the connection information is passed back to the requesting Control to pass to the requesting Manager, or it fails.

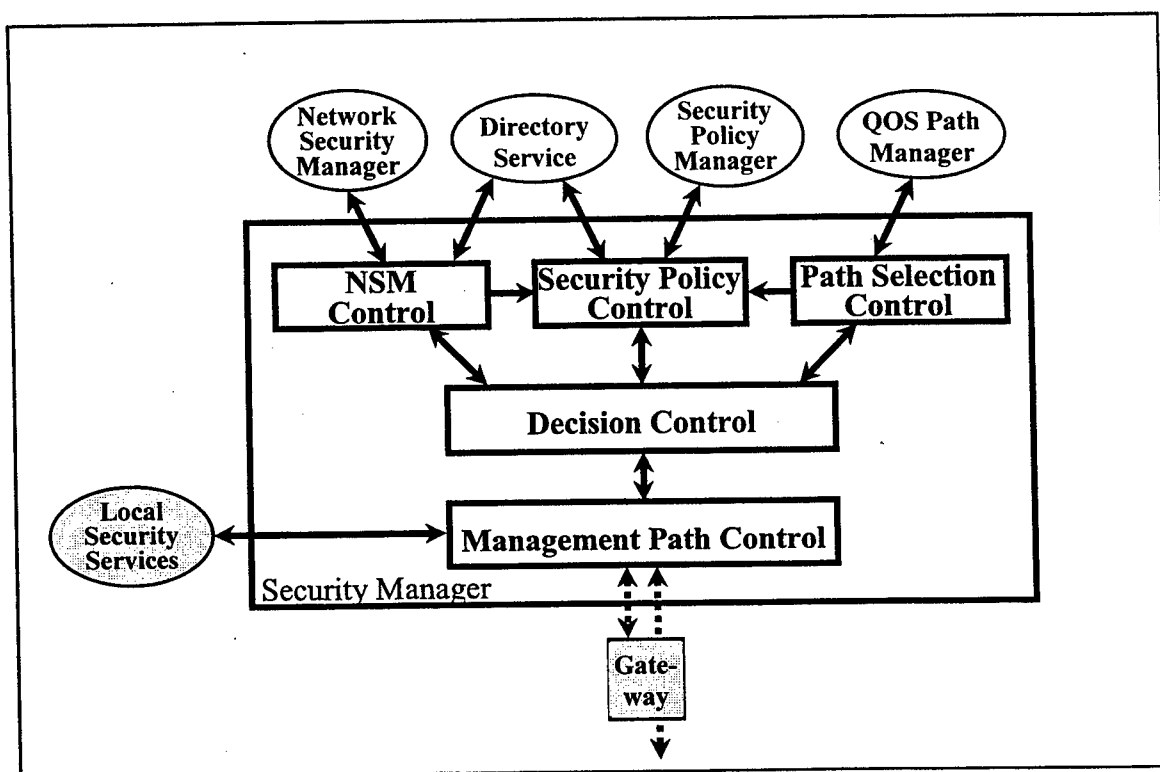


Figure 11. Security Manager Management Plane

The Management Path Control is also responsible for setting up the proper security context for the connection established by the QoS Path Manager and the Network Security Manager. The security policy for these management connections is administered separately from the policy in the Security Policy Server. This policy should be very basic, simple, easy to administer, and not easily changed. The Management Path Control is also responsible for applying the required security services for the connections established by the QoS Path Manager and the Network Security Manager.

The gateway devices that guard an enclave's boundary form an essential part of the security posture of the enclave. Each gateway device has a security policy that defines what the gateway allows to pass through it. At a minimum, these policies must be consistent with the overall enclave policy, as failure to support the enclave policy can result in a breach of enclave security. Ideally, the gateway policies should flow directly from the Security Policy Manager's security policy, ensuring consistency of the overall security policy. If a gateway device has a security management interface, then the Security Manager can use this interface to download the correct security policy into the gateway, ensuring the consistency of the security policies. Even more can be achieved by coordinating the actions of the Security Management Workstation with the gateway. For example, the Security Manager can configure the gateway to allow a specific connection to pass only when there is an actual need for the connection. The passageway through the gateway can be opened when the application needs it (i.e., as a result of the application making a connection request) and closed when the application no longer needs it (i.e., as a result of the application closing the connection.) Dynamic configuration of gateway policies also allows for a more efficient use of gateway resources, as resources can be allocated to applications that are actually using them instead of being held in reserve for applications that might need them. The development of standard API's and protocols for management of gateway devices would greatly enhance this approach.

The Management Path Control is responsible for interacting with gateway devices that have management interfaces. These connections from the SMW to a gateway require, at minimum, strong mutual authentication and integrity checking.

3.4.2 Security Policy Manager

The Security Policy Manager is responsible for storing the enclave security policy and securely providing policy information to the Security Manager. This server provides the security policy to the Security Manager so that it can make and manage ATM connections in accordance with the enclave security policy.

3.4.3 Network Security Manager

The Network Security Manager is responsible for providing information about the security characteristics and status of network resources (links, cryptographic devices, etc), and providing other network security management functions for the enclave. The network security manager gets its information from the network management system, other network security managers, and possibly other sources.

3.4.4 QoS Path Manager

The QoS Path Manager is responsible for finding a set of paths that meet the requested quality of service for a connection. The QoS Path Manager will obtain the information it needs from the topology management subsystem of the network management system. The topology manager will have an information base regarding the available links and the

current QoS parameters on each of them. The QoS Path Manager will form a list of potential paths, arrange the paths based on their appropriateness in satisfying the QoS, and return them to the Path Selection Control.

One of the goals of our research is to allow the Security Manager to have some control over the selection of the path during the connection setup phase. However, current ATM practice and standards provide the end-user application with limited control over the path selection and it is unclear to what degree this will be supported in future ATM standards. There are, however, some solutions to this problem:

1. Using PVCs: One approach to achieving control of the path selection is to use Permanent Virtual Circuits (PVCs). Many commercial network management systems can help in remotely establishing ATM PVCs between switches (e.g., IBM NetView has software called ATM Campus Manager to help in setting and managing the ATM connection). The PVCs can be set up in advance and used as needed, according to our QoS and Security requirements. If more paths are needed, they can be established with the help of a network management system. This approach seems to be a practical approach if one considers the currently available capabilities and standards in ATM networks.
2. Using SVCs: An alternative method is to use Switched Virtual Circuits (SVCs). The Private NNI standard provides the capabilities for specifying the selected switches in its connection setup message. The P-NNI specifies certain Designated Transit Lists (DTLs) that list switches through which the connection must pass. During the connection setup phase, intermediate switches examine the DTL list, use it to determine the next switch, and then pass the SETUP message along to the next identified switch and so on. The main problem with the current standards and their implementation is that the switch, and not the end-user system, controls the path selection. This means that some of the Security Manager functions need to be implemented in the ATM switch. Another problem with this approach is that current standards limit its application to private networks since DTLs cannot be passed across the Public NNI.
3. Hybrid: A hybrid solution, using both PVCs and SVCs, is also possible. The PVCs can be used within public networks and can be set up according to our QoS and Security requirements. SVCs can be used with private networks connected to these public networks. This hybrid solution provides the desired control over routing through the public network and provides the increased flexibility provided by SVCs within the private networks.
4. Public Networks: It may also be possible to use SVCs with public networks in a limited way: First, a SVC to the desired endpoint is set up through the public network, without regard to how the circuit is routed. Once the SVC is established, the network can be queried to obtain the final route used by the network. If this route is satisfactory (i.e., it meets the applicable security policy), the SVC can be used. If the route is not acceptable, the circuit is rejected. In this case, another SVC can be

established in the hope that the resulting path will be more favorable. Unfortunately, there is no guarantee that the new path will be better than the original and this lack of guarantee may limit the usefulness of this approach.

3.4.5 Directory Services

The Directory Services component of the Security Management Workstation stores information about network entities and resources for use by the Security Manager. When the Network Security Manager acquires information regarding the security characteristics and status of network resources from the network management system or other network security managers, it stores that information in the Directory Services component, which acts as a network security information repository.

3.4.6 Local Security Services

The Security Management Workstation has a set of security services (supported by either hardware or software mechanisms) that are used by the Security Manager. These security services must support secure communications with all the entities with whom the Security Manager needs to communicate securely. The Local Services component provides the interface to these security services; the SMW Interface uses this interface to apply any needed security services.

3.5 Gateways

Gateways are devices placed at enclave boundaries that control the flow of information into and out of the enclave. There are three basic types of gateways, each providing protection to the enclave in a different way:

Firewalls. A firewall prevents unauthorized access to the resources of the enclave by filtering out unauthorized incoming network traffic from outside the enclave. Only network traffic that results from actions initiated from inside the firewall or that is explicitly authorized is allowed to pass from outside the enclave to the inside.

Guards. A guard prevents unauthorized transmission of sensitive data from inside the firewall to outside the firewall by filtering outgoing network traffic. The guard allows only network traffic that meets the criteria established by the guard to pass through the filter.

Cryptographic Devices. Cryptographic devices, such as FASTLANE and TACLANE, encrypt and decrypt all network traffic leaving and entering the enclave. Properly managed cryptography ensures the authenticity and the privacy of the encrypted communications.

3.6 Network Management

The Network Management component represents a network management entity. The ATM USA uses this component in two ways. The Network Management component may provide information on network topology and capacity to the QoS Path Manager and may provide information on the network to the Network Security Manager.

Network topology information is maintained in a topology database MIB created by a service of the network management system. For example, commercial network management systems that provide topology discovery services use methods based on ping, traceroute, and ICMP echoes. The approach used to discover topologies varies from one network management system to another. For example in IBM NetView6000, there is a graphical interface to describe the network and the systems in it. By using that graphical user interface, one can obtain up-to-date information about the resources managed by the network management system. The Security Manager will use topology discovery services of the network management system to obtain network topology information.

4 Interfaces

As discussed in Section 3.2, several different components make up an enclave's security architecture: the application itself, the application workstation's Connection Manager, the enclave Security Manager, and the enclave gateway device. These components must be able to exchange information between one another. In particular, the application communicates with the Connection Manager, which communicates with the Security Manager, which in turn communicates with the enclave gateway device. Each of these three sets of communication exchange – application to Connection Manager, Connection Manager to Security Manager, and Security Manager to gateway device – is mediated by an interface consisting of a number of call primitives that may be used in passing commands or exchanging information. These three interfaces are discussed in turn in Sections 4.1, 4.2 and 4.4 below. A fourth possible interface, between the security managers of different enclaves, is discussed in Section 4.6. Finally, we discuss the coordination of calls across these interfaces in the context of establishing and terminating different types of ATM connections in Sections 4.3 and 4.5.

4.1 User Security Interface

In this section, we discuss the ATM USA user security interface at length. The relevant interface primitives are `CM_set_security_attributes` and `CM_query_security_attributes`, which provide the means for exchange between the application and the Connection Manager regarding ATM security.

The ATM USA User Security Application Programming Interfaces (APIs) collectively define the interface between user applications and the ATM user security services provided by the ATM USA. The following five goals were used to drive the design of the APIs described in this document.

- **Support for both security-aware and unaware user applications.** The APIs should be designed to support security-aware applications and should provide adequate default security for security-unaware applications.
- **User friendliness.** The APIs should be easy for the application programmer to use.
- **Consistency with other user services.** The APIs should provide security services in a manner that is consistent with how the application programmer obtains network services.
- **Minimize impact on user applications.** The APIs should have minimal impact on the design and implementation of applications. For example, security-unaware applications should at most require relinking. Security-aware applications will need additional code to use the security APIs, but should not require significant redesign.
- **Based on industry standards.** The APIs should be based on commonly accepted industry standards.

In keeping with the above design goals, this document describes two approaches to the design of the ATM User Security APIs. Based on the "Native ATM Services: Semantic Description Version 1.0," produced by the ATM Forum [13], the first approach provides a semantic definition of the ATM services available to applications from ATM networks. This semantic description defines a state machine, network primitives, and other information required for accessing native ATM services in a programming language-independent manner. Section 4.1.1 of this document describes this specification and extensions to it for the ATM USA.

The second approach is based on the Windows Socket 2 API. The ATM Forum semantic description is intended as the basis from which to derive language specific APIs. At least two sets of APIs, X/Open's XTI and X/Sockets APIs and the Windows Sockets 2 API have been recognized by the ATM Forum as providing legitimate syntax mappings of its semantic description [14, 15]. Both of these sets of APIs are derived from the BSD UNIX sockets implementation, but have been extended to support transport mechanisms other than TCP/IP. Section 4.1.2 of this document describes the Windows Sockets 2 API and extensions to it for the ATM USA.

4.1.1 ATM Native Security Services

This section presents User Security Services that are based on the semantic model defined by the ATM Forum. Section 4.1.1 presents an overview of the "Native ATM Services: Semantic Description Version 1.0" [13]. Section 4.1.1.2 presents the ATM User Security Services additions to this model.

4.1.1.1 The ATM Forum Specification

The ATM Forum defined a reference model for Native ATM Services. This model is shown in Figure 12.

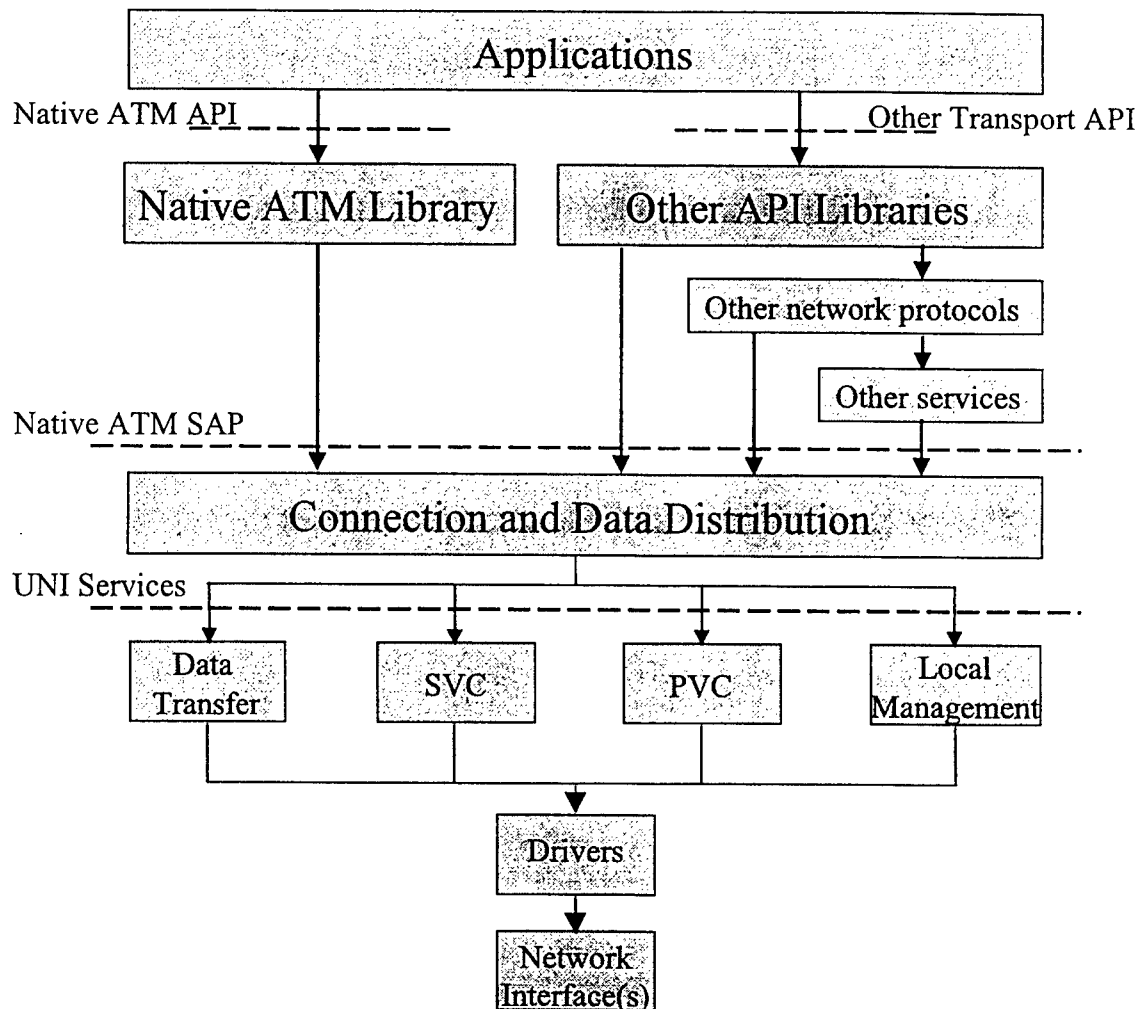


Figure 12. Native ATM Services Reference Model

The Native ATM API is the interface through which user applications both obtain access to and use services provided by an ATM network. The sequence of events that are required to set up, use, and shut down an ATM connection is described using the network state diagram shown in Figure 13. The left side of this diagram shows the states required to initiate and to use outgoing ATM point-to-point and point-to-multipoint connections. The right side of this diagram shows the states required to accept and to use an incoming ATM point-to-point or point-to-multipoint connection. In each case, the actions necessary to cause a transition from one state to the next state are shown. The actions are either *requests* generated by the application or *indications* or *confirmations* generated by the network. Note that this diagram does not show all possible actions. For example, the diagram does not show those actions that do not cause a state change.

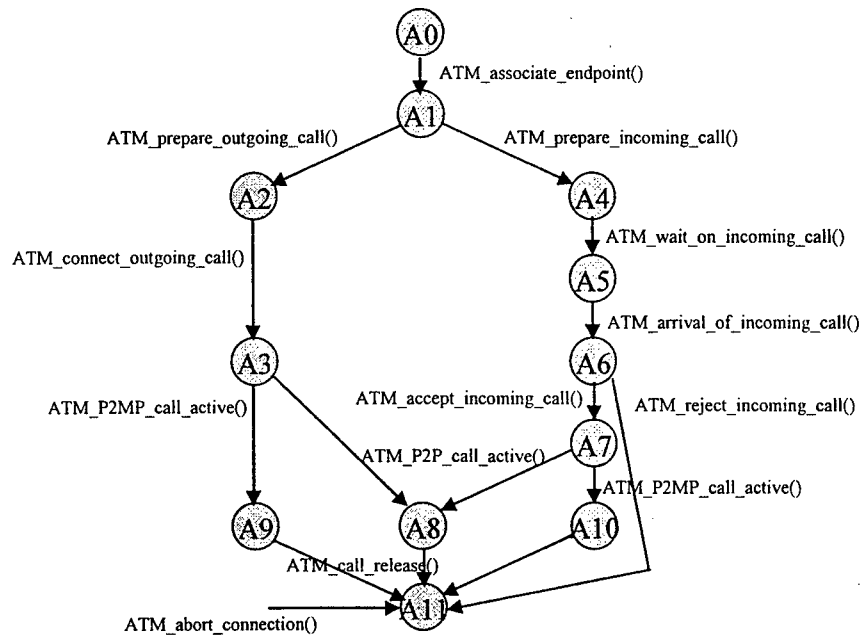


Figure 13. ATM Native Services State Diagram

Applications can use connections in states A8, A9, and A10 to send and receive data. Two primitives, `ATM_send_data()` and `ATM_receive_data()`, are provided for this purpose. Applications can query and set the attributes of connections when the connections are in state A2 or A6, for outgoing and incoming calls, respectively. The primitives, `ATM_query_connection_attributes()` and `ATM_set_connection_attributes()` are provided for this purpose. Figure 14 shows these primitives; they do not cause a state change. The ATM Forum Native Services: Semantic Description Version 1.0 lists the attributes that can be set for a connection [13].

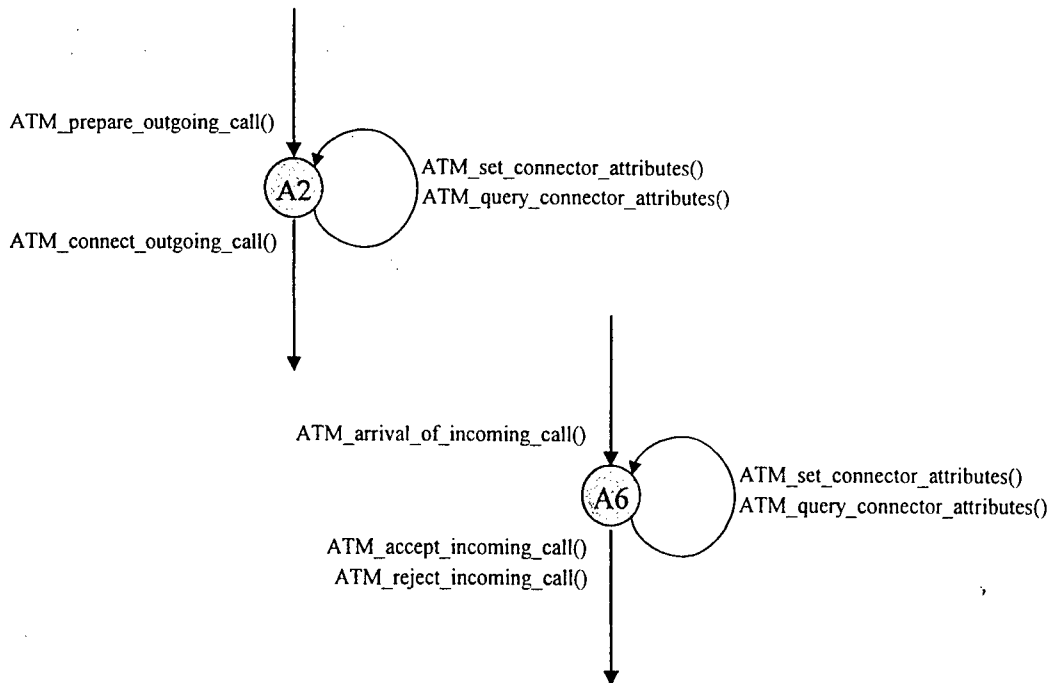


Figure 14. Setting Connection Attributes

4.1.1.2 ATM USA Security Extensions

The ATM USA User Security API extends the ATM Native Services API to provide user security services to user applications. The ATM USA approach is to augment the Native ATM API by adding a small number of security function calls that perform the necessary security functions. This approach has the advantage that both the syntax and the non-security semantics of the existing Native ATM calls are unchanged. Some of the Native ATM calls may have security-related side-effects, but these are transparent to the application.

The implementation of the ATM USA User Security API places the ATM USA Connection Manager between the application and the Native ATM Library that implements the Native ATM API, as shown in Figure 15. The Connection Manager implements a security-enhanced interface. The security-enhanced interface implements each primitive defined for the Native ATM API. The Connection Manager passes application calls unchanged to the Native ATM Library and passes indications and confirmations from the Native ATM Library to the application. For each primitive, the Connection Manager may or may not take additional security-related action, but this action will be transparent to the application. The enhanced interface also supports security-related calls to the Connection Manager. The Connection Manager processes these calls as part of its normal functioning.

Other transport libraries (e.g., classical IP over ATM) for accessing ATM services are handled similarly.

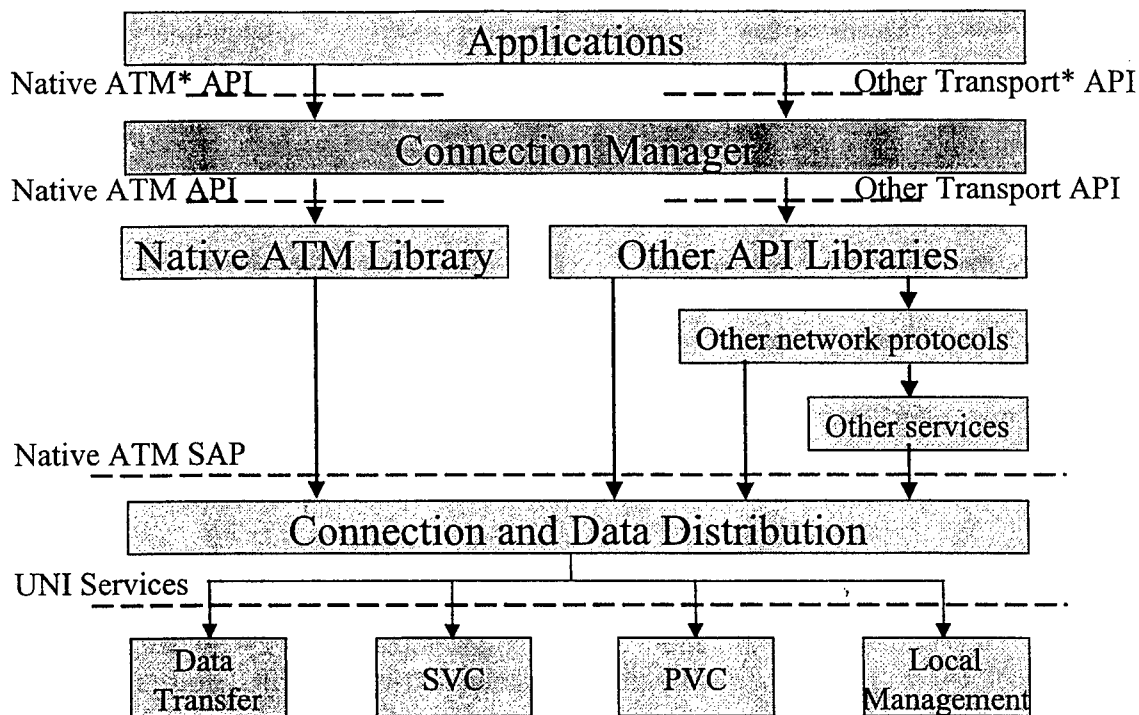


Figure 15. Security Reference Model

The Connection Manager supports two security-related function calls, `ATM_set_security_attributes()` and `ATM_query_security_attributes()`, which a security-aware application can use to set and query security parameters for a call. The Connection Manager provides default values for all security parameters for all connections for applications that do not set their own security parameters (such as security-unaware applications). If an application wishes to set security parameters for a connection, it must do so while the connection is in state A1, as shown in Figure 16. If the default values of the security parameters are acceptable to the application, it is not necessary for the application to set the values of any security parameter and the application can move directly to state 2 or state 4. The application may query the security parameters in any of the states A1 through A10.

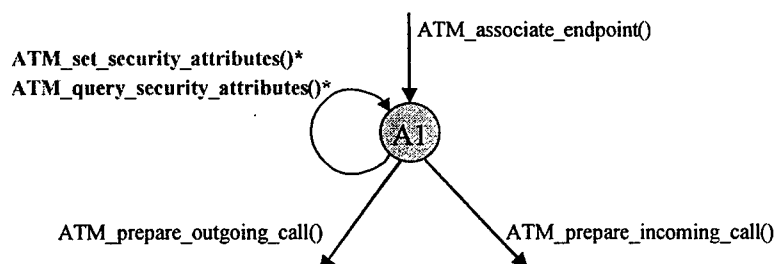


Figure 16. Setting Security Attributes

Note that the Connection Manager may restrict the security parameters that the application may set and query according to the security policy enforced by the Security Manager.

Section 4.1.3 discusses security attributes in more detail.

4.1.2 Windows Sockets

Windows Sockets 2 (WinSock2) is a Windows Open Services Architecture component consisting of an application programming interface (API) used by applications and service provider interfaces implemented by service providers. The WinSock2 API serves as an extensible abstraction layer providing Windows application programmers access to underlying services [16-18].

This section presents the syntactic mapping relating ATM Native Services to their associated WinSock2 API calls. A Windows application programmer wishing to specify ATM transport services for a given application would instead specify the associated WinSock2 calls, which serve as the high-level instantiations of the underlying ATM Native Services. For the ATM User Security Services primitives, `ATM_set_security_attributes` and `ATM_query_security_attributes`, there are no corresponding WinSock2 API calls and new ones must be introduced. Section 4.1.2.1 presents an overview of the Windows Sockets 2 API and its mapping to the ATM Native Services primitives. Section 4.1.2.2 presents the extensions to the WinSock2 API corresponding to the ATM User Security Services primitives.

4.1.2.1 The Windows Socket 2 API Specification

The ATM Forum has approved the WinSock2 API as a valid syntactic mapping of the semantic interface described in Section 4.1.1. Table 1 contains a summary of this mapping. For each ATM Native Services primitive in the left column, the right column shows the WinSock2 equivalent. For details, the reader is referred to the mapping of the Native ATM Services to WinSock2 [14].

Table 1. Windows Sockets 2 Mappings

ATM Native Services	Windows Sockets 2
ATM_associate_endpoint()	WSASocket()
ATM_prepare_outgoing_call()	bind() WSAIocctl()
ATM_prepare_incoming_call()	bind() listen()
ATM_wait_on_incoming_call()	WSAAccept() WSAEventSelect() with FD_ACCEPT
ATM_arrival_of_incoming_call()	WSAAccept() FD_ACCEPT
ATM_accept_incoming_call()	WSAAccept()
ATM_reject_incoming_call()	WSAAccept()
ATM_P2P_call_active()	FD_CONNECT (listening only)
ATM_P2MP-call_active()	FD_CONNECT (listening only)
ATM_call_release()	WSASendDisconnect()
ATM_abort_connection()	WSASendDisconnect() closesocket()
ATM_query_connection_attributes()	WSAIocctl() getsockopt()
ATM_set_connection_attributes()	WSAIocctl() setsockopt()
ATM_send_data()	WSASend()
ATM_receive_data()	WSARecv()
AMT_add_party()	WSAJoinLeaf()
ATM_drop_party()	WSASendDisconnect() Closesocket()
ATM_add_party_success	FD_CONNECT
ATM_add_party_reject	FD_CONNECT

4.1.2.2 ATM USA Security Extensions

The ATM USA API, the ATM USA extension to the WinSock2 API, introduces two new WinSock2 function calls, CMSetsockopt() and CMGetsockopt(), corresponding to the ATM User Security Services ATM_set_security_attributes and ATM_query_security_attributes primitives. These calls are used to set and query the security parameters for a socket (e.g., connection). Security parameters can be set for a socket

once the socket has been created with `WSASocket()` and before the socket is bound with `bind()` or `listen()`. Security parameters can be queried anytime after the socket is created with `WSASocket()` and before the socket is closed with `closesocket()`. More information about the relevant security parameters (represented as security options to be set or retrieved by the ATM USA API function calls) may be found in Section 4.1.3. The ATM USA API function calls have the following semantics:

4.1.2.2.1 CMSetsockopt()

Description Set a socket security option corresponding to the ATM USA Connection Manager security attributes.

int CMSetsockopt(

IN **SOCKET** *s*,
IN **int** *optname*,
IN **const char FAR*** *optval*,
IN **int** *optlen*

);

s A descriptor identifying a socket.
optname The socket option for which the value is to be set.
optval A pointer to the buffer in which the value for the requested option is supplied.
optlen The size of the *optval* buffer.

Remarks **CMSetsockopt()** sets the current value for an ATM USA Connection Manager-related socket option associated with an appropriate socket.

The following options, corresponding to parameters for the `ATM_set_security_attributes` primitive, are supported for **CMSetsockopt()**.

Table 2. CMSetsockopt and CMGetsockopt Parameters

Optname	Meaning
SourceID	String of bytes specifying the source ID.
DestinationID	String of bytes specifying the destination ID.
ConfidentialityService	Confidentiality enabled/disabled.
ConfidentialityScope	Range of application of confidentiality service.
ConfidentialityOptions	Confidentiality service option.
ConfidentialityAlgorithms	Confidentiality service algorithms.
IntegrityService	Integrity enabled/disabled.
IntegrityScope	Range of application of integrity service.
IntegrityOptions	Integrity service option.
IntegrityAlgorithms	Integrity service algorithms.
DataAuthenticationService	Authentication enabled/disabled.
DataAuthenticationScope	Range of application of authentication service.
DataAuthenticationOptions	Authentication service option.
DataAuthenticationAlgorithms	Authentication service algorithms.
KeyExchangeService	Key exchange enabled/disabled.
KeyExchangeScope	Range of application of key exchange service.
KeyExchangeOptions	Key exchange service option.
KeyExchangeAlgorithms	Key exchange service algorithms.
CertificateExchangeService	Certificate exchange enabled/disabled.
CertificateExchangeScope	Range of application of certificate exchange service.
CertificateExchangeOptions	Certificate exchange service option.
CertificateExchangeAlgorithms	Certificate exchange service algorithms.
KeyUpdateService	Key update enabled/disabled.
KeyUpdateScope	Range of application of key update service.
KeyUpdateOptions	Key update service option.
KeyUpdateAlgorithms	Key update service algorithms.
AccessControlService	Access control enabled/disabled.
AccessControlScope	Range of application of access control service.
AccessControlOptions	Access control service option.
AccessControlAlgorithms	Access control service algorithms.
LinkRestrictions	Link restrictions
DomainRestrictions	Domain restrictions

4.1.2.2.2 CMGetsockopt()

Description Retrieve a socket security option corresponding to the ATM USA Connection Manager security attributes.

```
int CMGetsockopt(  
    IN          SOCKET      s,  
    IN          int         optname,  
    OUT const char FAR*     optval,  
    IN OUT      int         optlen  
);
```

s A descriptor identifying a socket.
optname The socket option for which the value is to be retrieved.
optval A pointer to the buffer in which the value for the requested option is to be returned.
optlen A pointer to the size of the *optval* buffer.

Remarks **CMSetsockopt()** retrieves the current value for a ATM USA Connection Manager related socket option associated with an appropriate socket. The value associated with the selected option is returned in the buffer *optval*. The integer pointed to by *optlen* should originally contain the size of the buffer; on return, it will be set to the size of the value returned. If the option was never set with **CMSetsockopt()**, then **CMGetSockopt()** returns the default value for the option.

CMGetsockopt() supports the options shown in Table 2.

4.1.2.3 Windows and the ATM USA

ATM USA API security function calls and Winsock2 API calls made by Windows application programmers must be passed to the Connection Manager before being passed to the ATM transport layer in order for the ATM USA Connection Manager to function correctly.

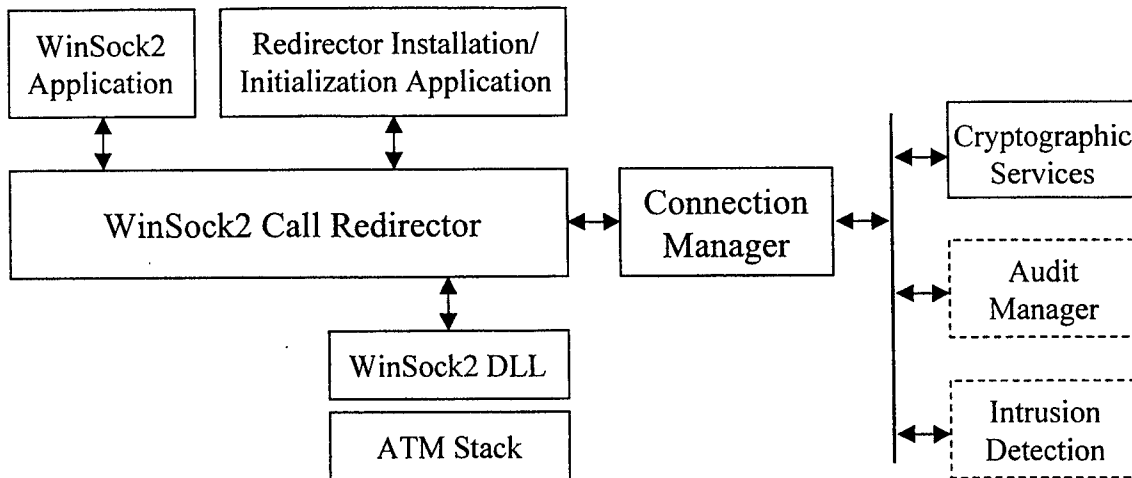


Figure 17. WinSock2 Generic Architecture

Figure 17 shows a generic architecture diagram relating Windows applications to the underlying ATM USA. The WinSock2 Services Architecture involves WinSock2 calls made by Windows applications being passed directly to the WinSock2 dynamic linked libraries (WinSock2 DLL). An extension to this architecture introduces a WinSock2 Call Redirector, which intercepts WinSock2 application calls prior to their entering the WinSock2 DLL and then hands them to various specialized DLL-based software components ("plug-ins") that then implement a particular functionality. A given plug-in may allow, modify or reject a particular WinSock2 call according to its own set of internal rules. Such a scheme allows for the transparent extension of WinSock2 DLL and application capabilities at the transport layer. The STARDUSTTM WinSock Component Architecture is one current instantiation of such an approach [19].

Building upon this approach, we may treat the ATM USA Connection Manager as a specialized security-related plug-in which acts upon WinSock2 calls that are passed to it by the WinSock2 Call Redirector in accordance with its security policy. Given the mapping between ATM Native Services primitives and the WinSock2 calls, along with their extensions for setting and querying security parameters, the substitution of the ATM USA Connection Manager for standard WinSock2 DLL-based plug-ins should be straightforward. The Connection Manager thus serves as a security shim, intercepting and applying security to WinSock2 calls.

This architecture allows the Connection Manager to be used as a front-end for a variety of security-related modules. For example, in Figure 17 the Connection Manager fronts three modules: a cryptographic services module, an audit module, and an intrusion detection module. The cryptographic services module supports the Connection Manager when the Connection Manager provides cryptographic services for a connection as part of its ATM USA role. The audit and intrusion detection modules do not play a role in the ATM USA architecture, but are included here to illustrate how the Connection Manager architecture can be extended to include other security related functionality. The advantage of this approach is that the Connection Manager needs to be inserted into the network protocol stack only once and then can be used to support other functionality.

The alternative approach is to have all these security services attempting to intercept the same resources at the same time and possibly interfering with each other. Assuring the correct functioning of each service individually would be much more difficult and expensive.

4.1.3 ATM USA Security Parameters

This section outlines the Security parameters used by the Connection Manager and the Security Manager to control the security of ATM connections.

As outlined in Section 4.1.1.2 for the ATM Native Services API, the Connection Manager supports two security-related function calls, `ATM_set_security_attributes()` and `ATM_query_security_attributes()`, which a security-aware application can use to set and query security parameters for a call. As outlined in Section 4.1.2.2 for the WinSock2 API, a Windows application programmer, setting or querying security for underlying ATM transport, would invoke the WinSock2 API extension calls `CMSetsockopt()` or `CMGetsockopt()` to set or query a specific ATM security option value. These options correspond exactly to the security parameters at the ATM USA layer.

In principle, a given application may set or query all of the possible security parameters. However, in practice, the enclave security policy specified by the Security Manager restricts those parameters that a given application can query or set to a subset of the whole. For instance, while link and domain restrictions may in principle be set by an application, the enclave security policy would not typically permit this, since an application would not normally possess the relevant network information to set these properties.

The various security properties and associated values play a necessarily double role in ATM security. On the one hand, they form the component elements of the enclave security policy specified by the Security Manager. That is, security policy statements, such as a statement allowing a connection if confidentiality services are enabled, are expressed in terms of the security properties and their values. On the other hand, in determining the permission for a specific connection request, the Security Manager must determine the actual values for the security properties pertaining to that connection. Values for properties associated with routing will come primarily from the network itself, while values for properties associated with security services may come from the application, the Connection Manager or the gateway, depending on where a given service is provided.

The ATM USA Security Parameters are divided into three categories: Identification, Security Services, and Routing. These categories cover all the necessary information required by the Connection Manager and Security Manager in making a connection request determination. The Identification category includes sections covering source and destination parameters. The Security Services category includes sections on confidentiality, integrity, authentication, key exchange, certificate exchange, key update and access control services. Each section in turn consists of the following parameters:

service enabled/disabled, range of operation, options, and algorithms. Finally, the Routing category includes a section on routing preferences, which covers link restriction and domain restriction parameters.

4.1.3.1 Identification

The SourceID and DestinationID parameters shown below in Table 3 are associated with source and destination identification.

Table 3. Identification Parameters

Parameter	Parameter Options
SourceID	ID (String of bytes)
DestinationID	ID (String of bytes)

4.1.3.2 Security Services

The Security Services parameters below cover confidentiality, integrity, authentication, key exchange, certificate exchange, key update and access control services. These services correspond to those specified in the ATM Forum "ATM Security Specification, Version 1.0" [12, 20]. The choice of these security services is driven by the desire to maintain compatibility with the ATM Forum security specification effort. The goal is for the ATM USA to support the security features outlined by the ATM Forum. However, ATM USA supports security services beyond the needs of the commercial sector, meeting the requirements of the Department of Defense. The Security Services parameters defined below meet these goals.

Each security service has associated parameters: The Service parameter for each security service specifies whether or not that service is enabled. The Scope parameter specifies the scope over which the service is applied. The possible values are application-end-to-end (the service is applied over the entire path, from source application to destination application), connection-manager-end-to-end (the service is applied to the entire path from the source connection manager to destination connection manager), enclave-to-enclave (the service is applied from the source enclave to the destination enclave, but not within those enclaves themselves), and link-to-link (the service is applied over only the unprotected links in the path, and not over any intermediate enclaves). The Options parameter specifies whether that service is not supported, supported or required at the ATM Cell Level, as detailed in the ATM Security Specification. Finally, the Algorithms parameter specifies the possible algorithms that may be selected for each service. These possible algorithm values exactly match those given in the ATM Security Specification, although an additional value – a user defined algorithm – has been added for user flexibility.

Note that the suggested values in the following tables provide compatibility with the ATM Forum Security Specification. However, it is a simple matter to extend the possible values of these parameters to meet the specific needs of the Department of Defense.

4.1.3.2.1 Confidentiality

Confidentiality is a fundamental security service, protecting message privacy during transport. In the ATM Forum security specification, confidentiality is provided via ATM layer encryption on a per-virtual circuit basis, where encryption is applied to the sequence of data cells in a given virtual circuit. For further information, consult Sections 3.2.1 and 6.2.3.1 in the "ATM Security Specification" [12]. ATM USA also supports confidentiality services in other ways. For example, ATM USA allows confidentiality services to be applied at other layers in the protocol stack, and allows other confidentiality algorithms more appropriate to Department of Defense needs. Table 4 below lists the confidentiality parameters supported by the ATM USA.

Table 4. Confidentiality Parameters

Parameter	Parameter Options
ConfidentialityService	Enabled Disabled
ConfidentialityScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
ConfidentialityOptions	Not Supported Supported At ATM Cell Level Required At ATM Cell Level
ConfidentialityAlgorithm	DES with 56 bits effective key DES with 40 bits effective key Triple-DES with 112 bits effective key FEAL, N=32, 64 bit key, no key block parity User-defined data confidentiality algorithm
ConfidentialityMode	Unspecified CBC Counter Mode ECB User-defined data confidentiality mode of operation

4.1.3.2.2 Integrity

Integrity is a fundamental security service, protecting messages from alteration during transport. In the ATM security specification, integrity is provided on a per-virtual channel basis, where cryptographic checksums (also termed hashes, message authentication codes, or digests) are applied to the sequence of common-part service data units in a given virtual channel. For further information, consult Sections 3.3.1 and 6.2.3.2 in the "ATM Security Specification." ATM USA also supports integrity services in other ways. For example, ATM USA allows integrity services to be applied at other layers in the protocol stack, and allows other integrity algorithms that are more appropriate to Department of Defense needs. Table 5 below lists the integrity parameters supported by the ATM USA.

Table 5. Integrity Parameters

Parameter	Parameter Options
IntegrityService	Enabled Disabled
IntegrityScope	Application-end-to-end Connection-manager-end-to-end
IntegrityOptions	Not supported Supported With Replay/Reordering Protection Supported Without Replay/Reordering Protection Required With Replay/Reordering Protection Required Without Replay/Reordering Protection
IntegrityAlgorithm	HMAC-MD5 HMAC-SHA-1 HMAC-RIPEMD-160 MAC generated using DES in CBC mode MAC generated using DES-40 in CBC mode MAC generated using Triple-DES in CBC mode MAC generated using FEAL in CBC mode User-defined data integrity algorithm
ReplayProtection	No replay/reordering protection provided. Replay/reordering protection provided
HashAlgorithm	MD5 SHA-1 RIPEMD-160 User-defined hash algorithm

4.1.3.2.3 Authentication

Authentication is a fundamental security service, verifying the identities of entities engaged in secure ATM messaging. In the ATM security specification, authentication, whether bidirectional or unidirectional, is provided on a per-virtual circuit basis by means of digital signatures and message hashes. For further information, consult Sections 3.1.1 and 6.2.3.7 in the “ATM Security Specification.” ATM USA also supports authentication services in other ways. For example, ATM USA provides for authentication at different layers of the protocol stack and for authentication algorithms that are more appropriate to Department of Defense needs. Table 6 below lists the authentication parameters supported by the ATM USA.

Table 6. Authentication Parameters

Parameter	Parameter Options
AuthenticationService	Enabled Disabled
AuthenticationScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
AuthenticationOptions	Not supported Supports Authentication Requires Authentication
AuthenticationAlgorithms	RSA DSA Elliptic Curve/DSA like ESIGN DES/CBC DES40/CBC Triple DES/CBC FEAL/CBC User-defined data authentication algorithm

4.1.3.2.4 KeyExchange

Key exchange is a supporting security service. In enabling the public key exchange of a secret master key for subsequent symmetric key cryptography, it forms one aspect of ATM encryption services. For further information, consult Sections 5.2 and 6.2.3.5 in the "ATM Security Specification." ATM USA supports key exchange services in other ways as well. For example, ATM USA provides for key exchange at different layers of the protocol stack and for key exchange algorithms more appropriate to Department of Defense needs. Table 7 below lists the key exchange parameters supported by the ATM USA.

Table 7. Key Exchange Parameters

Parameter	Parameter Options
KeyExchangeService	Enabled Disabled
KeyExchangeScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
KeyExchangeOptions	Not supported Supports Key Exchange Requires Key Exchange
KeyExchangeAlgorithms	RSA Diffie-Hellman Elliptic Curve/Diffie-Hellman Analogue (prime field) Elliptic Curve/Diffie-Hellman Analogue (characteristic 2 field) DES/CBC DES40/CBC Triple DES/CBC FEAL/CBC User-defined key exchange algorithm

4.1.3.2.5 Certificate Exchange

Certificate exchange is a supporting security service. In enabling entities to verify public key bindings, it forms one aspect of ATM encryption services. For further information, consult Section 6.2.5.4 in the “ATM Security Specification.” ATM USA supports certificate exchange services in other ways as well. For example, ATM USA provides for certificate exchange at different layers of the protocol stack and for certificate exchange algorithms more appropriate to Department of Defense needs. Table 8 below lists the certificate exchange parameters supported by the ATM USA.

Table 8. Certificate Exchange Parameters

Parameter	Parameter Options
CertificateExchangeService	Enabled Disabled
CertificateExchangeScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
CertificateExchangeOptions	Not supported Supports Certificate Exchange Requires Certificate Exchange
CertificateExchangeCredential	X.509 Certificate V1 X.509 Certificate V2 X.509 Certificate V3 User-defined certificate exchange algorithm

4.1.3.2.6 Key Update

Key update is a supporting security service. In enabling the update of symmetric session keys from a pre-exchanged secret master key, it limits the amount of ciphertext exchanged with a given session key and, as such, forms one aspect of ATM encryption services. For further information, consult Sections 5.3 and 6.2.3.6 in the "ATM Security Specification." ATM USA supports key update services in other ways as well. For example, ATM USA provides for key update at different layers of the protocol stack and for key update algorithms more appropriate to Department of Defense needs. Table 9 below lists the key update parameters supported by the ATM USA.

Table 9. Key Update Parameters

Parameter	Parameter Options
KeyUpdateService	Enabled Disabled
KeyUpdateScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
KeyUpdateOptions	Supports Session Key Update Requires Session Key Update
KeyUpdateAlgorithms	SKE with MD5 SKE with SHA-1 SKE with RIPEMD-160 User-defined key update algorithm

4.1.3.2.7 Access Control

Access control is a fundamental security service, restricting ATM connections that violate a given security policy. In ATM security, access control is provided on a per-virtual circuit basis and is performed during connection establishment. For further information, consult Section 3.4 in the "ATM Security Specification." ATM USA supports access control services in other ways as well. For example, ATM USA provides for access control at different layers of the protocol stack and for access control methods more appropriate to Department of Defense needs. Table 10 below lists the access control parameters supported by the ATM USA.

Table 10. Access Control Parameters

Parameter	Parameter Options
AccessControlService	Enabled Disabled
AccessControlScope	Application-end-to-end Connection-manager-end-to-end Enclave-to-enclave Link-to-link
AccessControlOptions	Not supported Supports Access Control Requires Access Control
Label based Access Control	Hierarchical Levels Access Categories User-defined label-based access control
Role based Access Control	User-defined role-based access control
Identity based Access Control	User-defined identity-based access control

4.1.3.2.8 Routing

Network configuration parameters play a fundamental role in both the statement of security policy and the application of access control. In setting up a virtual circuit between two endpoints in an ATM network, various possible paths present themselves. These paths will vary in the types of security risk they present to ATM messaging. The following routing preferences enable the specification of required or restricted link and domain characteristics associated with a given path in the ATM security policy. Table 11 below lists the routing parameters supported by the ATM USA.

Table 11. Routing Parameters

Parameter	Parameter Options
PhysicalLinkRestrictions	[NO]Copper [NO]Fiber [NO]Wireless
GeographicalLinkRestrictions	[NO]CONUS [NO]NATO [NO]IRAQ ...
DomainRestrictions	[NO]CONUS [NO]NATO [NO]IRAQ ...

4.2 Client Workstation to Security Management Workstation Interface

The following calls, listed in Table 12, define the Connection Manager to Security Manager interface for the ATM USA. Various security-mediated operations associated with the establishment or release of a connection will involve signaling between the Connection Manager and the Security Manager. In the cases where the Connection Manager must query the Security Manager for permission, this signaling takes the form of a request/confirm pair. In the cases where permission for connection establishment has already been negotiated and granted by the Security Manager, signaling involves the Connection Manager informing the Security Manager of connection establishment success or failure without the Security Manager sending a confirm. The various operations, along with the associated Connection Manager to Security Manager signaling, are listed in Table 12 below.

Table 12. Connection Manager to Security Manager Interface Primitives

ATM Security-Mediated Connection Establishment/Release Operation	Connection Manager to Security Manager Interface Primitives
Making a Point-to-Point Call	SM_outgoing_call_request SM_outgoing_call_confirm SM_call_active
Accepting a Point-to-Point Call	SM_wait_on_incoming_call SM_arrival_incoming_call_request SM_arrival_incoming_call_confirm SM_accept_incoming_call_request SM_accept_incoming_call_confirm SM_call_active
Connection Release by an Application or the Network	SM_call_release
Adding a Party to a Point-to-Multipoint Connection, or a Leaf Initiated Join at the Root or Leaf Nodes	SM_add_party_request SM_add_party_confirm SM_add_party_success
Dropping a Party from a Multipoint Connection	SM_drop_party_request SM_drop_party_confirm SM_drop_party_success

4.2.1 SM_outgoing_call_request

In negotiating the connection establishment for an application making a point-to-point call, the Connection Manager passes the SM_outgoing_call_request primitive to the Security Manager, requesting permission for outgoing call connection establishment. The primitive carries the values of any application-dependent or connection-related security parameters to the Security Manager, which uses these in its security evaluation of the connection request.

4.2.2 SM_outgoing_call_confirm

In response to the Connection Manager's SM_outgoing_call_request, the Security Manager evaluates the connection, and replies to the Connection Manager with SM_outgoing_call_confirm, either permitting or denying the connection establishment. This primitive may also include security-relevant information related to its response.

4.2.3 SM_wait_on_incoming_call_request

In waiting for an incoming connection setup request from the network, the Connection Manager passes a SM_wait_on_incoming_call_request to the Security Manager, requesting that the Security Manager open a control channel to the network for the receipt of a SETUP control signal.

4.2.4 SM_arrival_incoming_call_request

In negotiating the connection establishment for an application accepting a point-to-point call, the Connection Manager first waits for an incoming call from the network. Upon receiving a SETUP control-plane message from the network, the Connection Manager passes a SM_arrival_incoming_call_request to the Security Manager, requesting permission for outgoing call connection establishment. This primitive carries the values of any application-dependent or connection-related security parameters to the Security Manager, which uses these in its security evaluation of the connection request.

4.2.5 SM_arrival_incoming_call_confirm

In response to the Connection Manager's SM_arrival_incoming_call_request, the Security Manager evaluates the connection, and replies to the Connection Manager with SM_arrival_incoming_call_confirm, either permitting or denying the connection establishment. This primitive may also include security-relevant information related to its response.

4.2.6 SM_accept_incoming_call_request

Following the permission granted by the Security Manager for the arrival of the incoming call, the Connection Manager informs the application of the incoming call. The application may in turn query or set relevant connection attributes, making it necessary for the Security Manager to reevaluate the establishment of the connection, since the application may alter connection attributes relevant to the enclave's security policy. Given this, the Connection Manager passes an SM_accept_incoming_call_request to the Security Manager, again requesting permission for outgoing call connection establishment. This primitive carries the values of any application-dependent or connection-related security parameters to the Security Manager, which uses these in its security evaluation of the connection request.

4.2.7 SM_accept_incoming_call_confirm

In response to the Connection Manager's SM_accept_incoming_call_request, the Security Manager evaluates the connection, and replies to the Connection Manager with SM_accept_incoming_call_confirm, either permitting or denying the connection establishment. This primitive may also include security-relevant information related to its response.

4.2.8 SM_call_active

For an application either making or accepting a point-to-point call, the Connection Manager passes SM_call_active to the Security Manager. This follows the Security Manager's permission for and the subsequent establishment of the ATM connection, and informs the Security Manager of its successful establishment.

4.2.9 SM_call_release

Upon release of a previously established single point connection, either by the application or by the network, the Connection Manager informs the Security Manager of the release by passing the primitive SM_call_release.

4.2.10 SM_add_party_request

Three distinct operations involve the SM_add_party_request primitive; adding a party to a point-to-multipoint connection at the root node, a leaf-initiated join at the root node, and a leaf initiated join at the leaf node. All three of these operations are similar, and, in the stage at which the Connection Manager to Security Manager signaling comes into play, they are in fact identical. Given this, the description of SM_add_party_request is identical for all three operations.

In negotiating the adding of a leaf to an existing point-to-multipoint connection, the Connection Manager passes the SM_add_party_request primitive to the Security Manager, requesting permission for connection establishment to the new leaf. The primitive carries the values of any application-dependent or connection-related security parameters to the Security Manager, which uses these in its security evaluation of the connection request.

4.2.11 SM_add_party_confirm

In response to the Connection Manager's SM_add_party_request, the Security Manager evaluates the connection, and replies to the Connection Manager with SM_add_party_confirm, either permitting or denying the connection establishment to the new leaf. This primitive may also include security-relevant information related to its response.

4.2.12 SM_add_party_success

Following the Security Manager's permission for and the subsequent establishment of the new leaf connection, the Connection Manager passes SM_add_party_success to the Security Manager, informing the Security Manager of the leaf connection's successful establishment.

4.2.13 SM_drop_party_request

In negotiating the dropping of a party from an existing multipoint connection, the Connection Manager passes the SM_drop_party_request primitive to the Security Manager, requesting permission to close a given point-to-multipoint connection node. The primitive carries the values of any application-dependent or connection-related security parameters to the Security Manager, which uses these in its security evaluation of the drop request.

4.2.14 SM_drop_party_confirm

In response to the Connection Manager's SM_drop_party_request, the Security Manager evaluates the connection, and replies to the Connection Manager with SM_drop_party_confirm, either permitting or denying the closing of the connection. This primitive may also include security-relevant information related to its response.

4.2.15 SM_drop_party_success

Following the Security Manager's permission for and the subsequent dropping of the connection, the Connection Manager passes SM_drop_party_success to the Security Manager, informing the Security Manager of the successful dropping of the node from the multipoint connection.

4.3 Security and Signaling Coordination

This section describes the role of the Connection Manager and the Security Manager in the procurement and use of ATM services from the network by an application. This section addresses the interactions of the Connection Manager with applications, underlying network services, and the Security Manager; and describes how and when these interactions occur. The primary purpose of the Connection Manager is to enforce the enclave security policy; the Security Manager controls and directs the Connection Manager in the Connection Manager's enforcement of the security policy.

The Connection Manager forms a shim between the application and the network, as shown in the Security Reference Model, Figure 18. As such, all operations involving the setup, use, and shut-down of ATM connections must pass through and negotiate with the Connection Manager. The Connection Manager must pass requests and responses from the application layer to the underlying ATM Adaptation Layer (AAL) and must pass confirmations and indications from the AAL to the application layer.¹ The Connection Manager may also take action, in addition to passing the operation from one layer to another. For example, the Connection Manager may change its internal state, modify the operation, or communicate with the Security Manager. The Connection Manager communicates with the Security Manager to seek advice on the enclave security policy or to inform the Security Manager of changes in connection status.

¹ The primitives *request*, *indicate*, *confirm*, and *response* are defined in the OSI Reference Model.

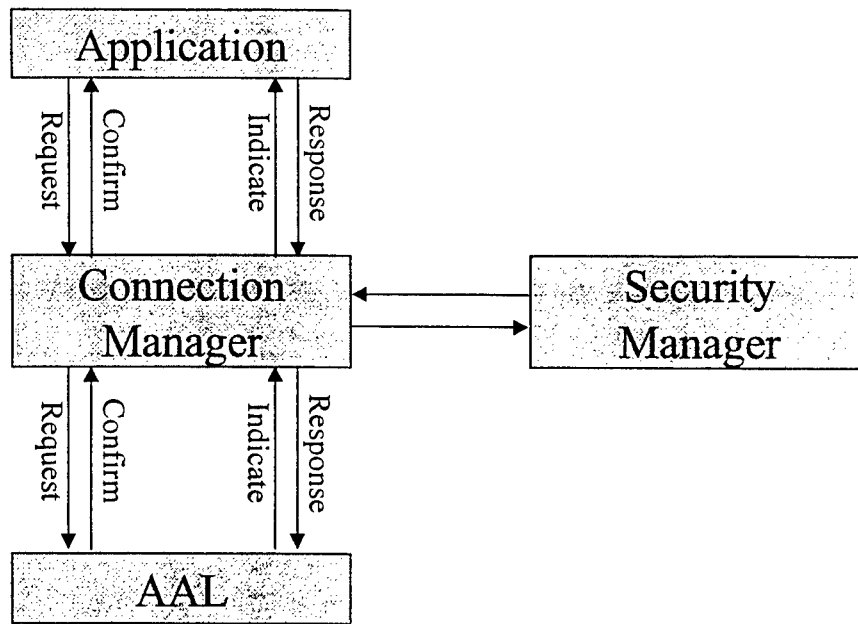


Figure 18. Network Security Shim Architecture

Figure 19 through Figure 32 show the ordering and coordination of calls between an application, the Connection Manager, the Security Manager and AAL for several different aspects of ATM signaling. In these figures, calls have been introduced as part of the ATM USA API security extensions are shown as bolded. We will discuss each of these figures in turn.

In the following diagrams, the Connection Manager processes every call made by the application. Some calls may cause a change in the internal state of the Connection Manager. Some calls may cause the Connection Manager to communicate with the Security Manager. The Connection Manager passes most calls to the AAL for further processing.

In these diagrams, the components and operations internal to the Connection Manager and the Security Manager, discussed at length in Sections 3.3 and 3.4, are not shown. In addition, the interface components internal to the Connection Manager and the Security Manager – specifically, the Application Control and the Security Manager Interface in the Connection Manager and the Connection Manager Interface and Management Path Control in the Security Manager – are not shown. Thus, although the security management calls outlined in Table 12 are passed between the Connection Manager’s Security Manager Interface and the Security Manager’s Connection Manager Interface, in the diagrams below, these calls are shown as passing between the Connection Manager and Security Manager proper. Similarly, the user security interface calls discussed in Section 4.1 are in fact passed between the application and the Connection Manager’s Application Control component, although the diagrams below show the Application passing user security interface calls with the Connection Manager proper.

4.3.1 Point-to-Point and Point-to-Multipoint Connection Setup

In Figure 19, the following sequence of steps are followed to establish a point-to-point connection:

1. The ATM_associate_endpoint request associates an API endpoint with the potential API connections that will use that endpoint. The application passes this request to the Connection Manager, and the Connection Manager passes it to the AAL.
2. If the application is security-aware, the application can use the ATM_set_security_attributes and ATM_query_security_attributes requests to set and query security parameters for the connection.
3. The ATM_prepare_outgoing_call request sets up the data structures that hold the characteristics of the outgoing connection.
4. The ATM_set_connection_attributes and ATM_query_connection_attributes requests may be used by an application to modify or obtain an attribute value of a connection prior to establishing the connection. These requests are passed first to the Connection Manager, and then by the Connection Manager to the AAL.
5. The ATM_connect_outgoing_call request initiates a connection across the ATM network. Here, it is initiated by the application, which passes it to the Connection Manager.
6. After receiving the connection initiation from the application, the Connection Manager interacts with the Security manager through the SM_outgoing_call_request and SM_outgoing_call_confirm messages. The Connection Manager uses the SM_outgoing_call_request message to pass security-relevant information about the connection to the Security Manager. The Security Manager responds to the Connection Manager with a SM_outgoing_call_confirm message containing the approval or denial of the connection request.
7. If the Security Manager approves the connection, the Connection Manager passes the ATM_connect_outgoing_call request to the AAL. The AAL requests the connection setup from the network, which in turn confirms the connection setup to the AAL.
8. Using the ATM_P2P_call_active confirmation, the AAL signals to the Connection Manager that the point-to-point connection is now in an active state.
9. Using the SM_call_active message, the Connection Manager informs the Security Manager that the connection is in an active state.
10. The Connection Manager passes the ATM_P2P_call_active confirmation back to the application, informing it of the connection's active state.

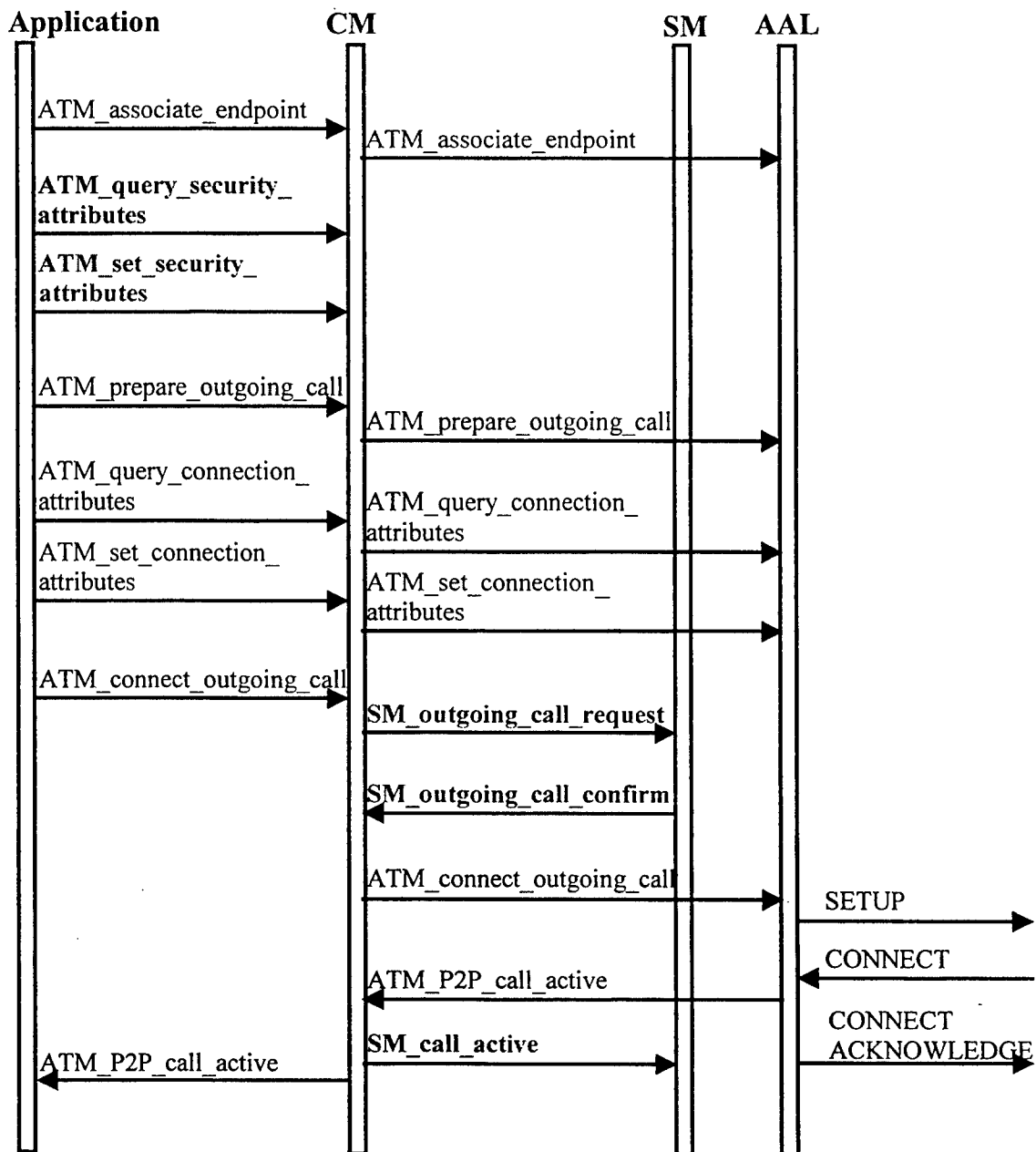


Figure 19. Making a Point-to-Point Call

Establishment of a point-to-multipoint connection from the root node of the connection is the same, except that in steps 8 and 10 the ATM_P2P_call_active confirmation is replaced by the ATM_P2MP_call_active confirmation.

If the Security Manager rejects the call, as shown in Figure 20, the connection attempt fails. In this case, the Connection Manager sends an ATM_call_release indication to the application. This is the same indication that the application receives when the network is unable to complete the call.

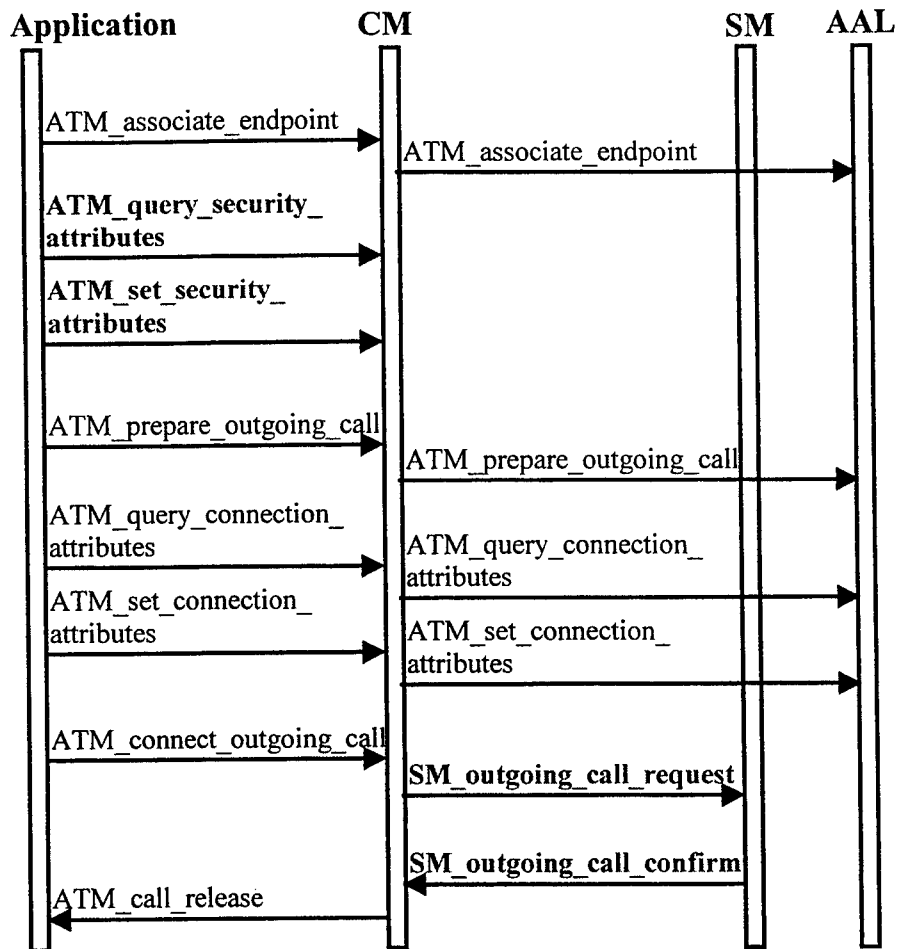


Figure 20. Security Failure for Point-to-Point Call

4.3.2 Point-to-Point and Point-to-Multipoint Connection Acceptance

In Figure 21, the following sequence of steps are followed to accept a point-to-point connection:

1. The ATM_associate_endpoint request associates an API endpoint with the potential API connections that will use that endpoint. The application passes this request to the Connection Manager, and the Connection Manager passes it to the AAL.
2. If the application is security-aware, it can use the ATM_set_security_attributes and ATM_query_security_attributes requests to set and query security parameters for the connection.

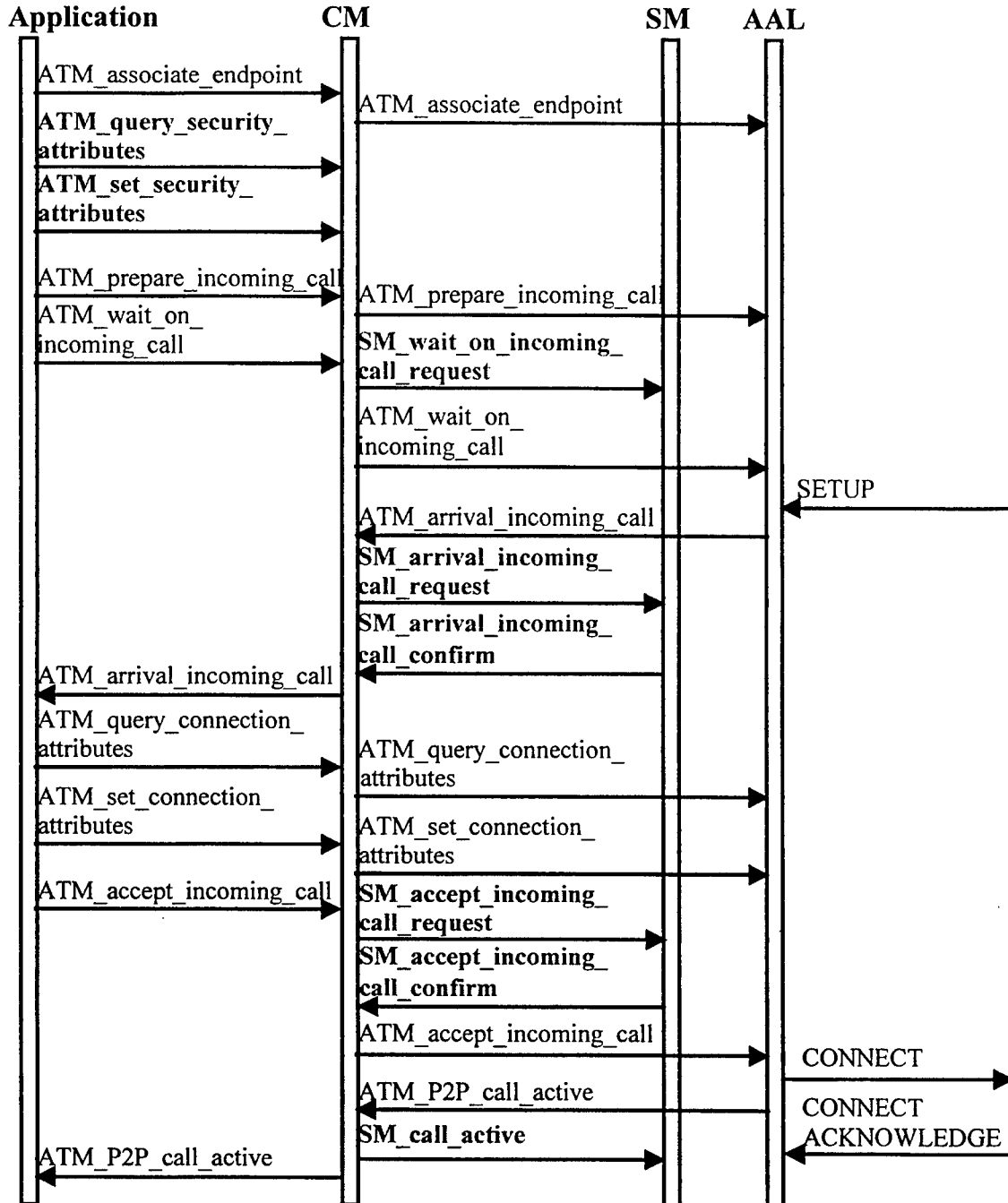


Figure 21. Accepting a Point-to-Point Call

3. The `ATM_prepare_incoming_call` request sets up the incoming call distribution tables, while the `ATM_wait_on_incoming_call` request activates the incoming call distribution function for this endpoint. These requests are passed first to the Connection Manager, and then by the Connection Manager to the AAL. On receiving the `ATM_wait_on_incoming_call` request, the Connection Manager notifies the Security Manager with a `SM_wait_on_incoming_call` message.

4. Upon the arrival of a SETUP indication from the network, the AAL passes the ATM_arrival_incoming_call indicator to the Connection Manager in order to notify it that an ATM call has arrived from the network.
5. After receiving the call notification from the AAL, the Connection Manager interacts with the Security Manager through the SM_arrival_incoming_call_request and SM_arrival_incoming_call_confirm messages. The Connection Manager uses the SM_arrival_incoming_call_request message to pass security-relevant information about the connection to the Security Manager. The Security Manager responds to the Connection Manager with a SM_arrival_incoming_call_confirm message containing the approval or denial of the connection request.
6. Following approval by the Security Manager, the Connection Manager passes the ATM_arrival_incoming_call indication on to the application.
7. After notification of the incoming call, the application may use the ATM_query_connection_attributes and ATM_set_connection_attributes requests to obtain or modify an attribute value of a connection prior to connection establishment. These requests are passed first to the Connection Manager, and then by the Connection Manager to the AAL.
8. Following this, the application accepts the incoming network call, passing the ATM_accept_incoming_call request to the Connection Manager.
9. After receiving the incoming call acceptance from the application, the Connection Manager interacts with the Security manager through the SM_accept_incoming_call_request and SM_accept_incoming_call_confirm messages. The Connection Manager uses the SM_accept_incoming_call_request message to pass security-relevant information about the connection to the Security Manager. The Security Manager responds to the Connection Manager with a SM_accept_incoming_call_confirm message containing the approval or denial of the connection request.
10. If the Security Manager approves the connection, the Connection Manager passes on the ATM_accept_incoming_call request to the AAL. The AAL confirms the connection setup request to the network.
11. Using the ATM_P2P_call_active indication, the AAL signals to the Connection Manager that the point-to-point connection is now in an active state.
12. Using the SM_call_active message, the Connection Manager informs the Security Manager that the connection is in an active state.
13. The Connection Manager passes the ATM_P2P_call_active indication back to the application, informing it of the connection's active state.

Acceptance of a point-to-multipoint connection by a leaf node is exactly the same, except that in step 11 and step 13 the ATM_P2P_call_active indication is replaced by the ATM_P2MP_call_active indication.

If the Security Manager rejects the incoming call in step 5, as shown in Figure 22, the connection attempt fails. In this case, the Connection Manager sends an ATM_reject_incoming_call response to the network. If the Security Manager rejects the incoming call in step 9, the response is similar. The connection attempt fails and the Connection Manager sends an ATM_reject_incoming_call response to the network. The application would make the same response if it were to reject the call.

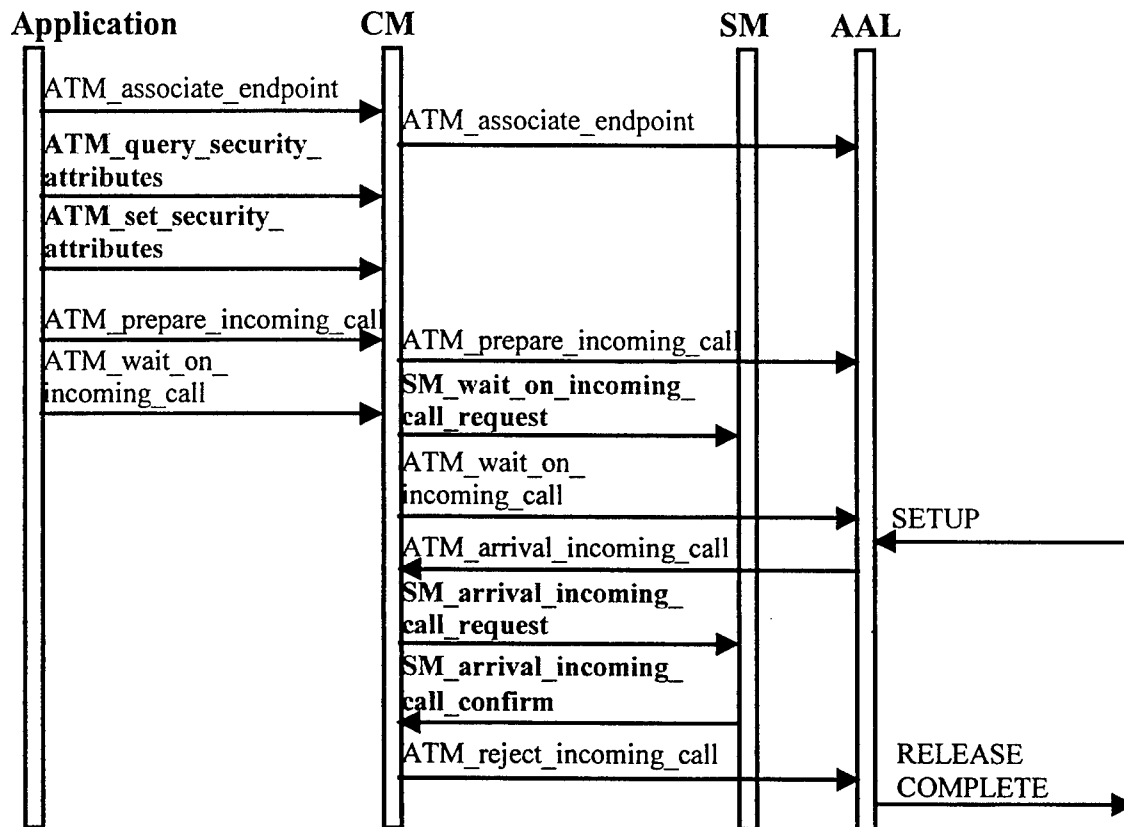


Figure 22. Security Rejection of Point-to-Point Call

4.3.3 Adding a Party to a Point-to-Multipoint Connection at the Root Node

In Figure 23, an application at the root node of a point-to-multipoint connection adds a party to the point-to-multipoint connection through the following sequence of steps:

1. The application passes the ATM_add_party request, which adds a leaf to an existing point-to-multipoint connection, to the Connection Manager.
2. The Connection Manager interacts with the Security manager through the SM_add_party_request and SM_add_party_confirm messages. The Connection Manager

passes security-relevant information about the connection to the Security Manager, which in turn responds to the Connection Manager with approval or denial of the “add party” request.

3. If the Security Manager approves the “add party” request, the Connection Manager passes on the ATM_add_party request to the AAL. The AAL requests the additional connection from the network, which in turn confirms the request.
4. Upon confirmation of the “add party” success or denial from the network, the AAL informs the Connection Manager by passing back either the ATM_add_party_success or the ATM_add_party_reject confirmation. The Connection Manager in turn passes the confirmation back to the application, informing it of the success or failure of the “add party” attempt. The Connection Manager also informs the Security Manager of the success or failure of the “add party” request using a SM_add_party_success or SM_add_party_failure message.

If the Security Manager rejects the addition of a leaf to a multiparty call, as shown in Figure 24, the connection attempt fails. In this case, the Connection Manager sends an ATM_add_party_reject confirmation to the application. This confirmation is the same confirmation that the application would receive if the network had been unable to add the leaf node.

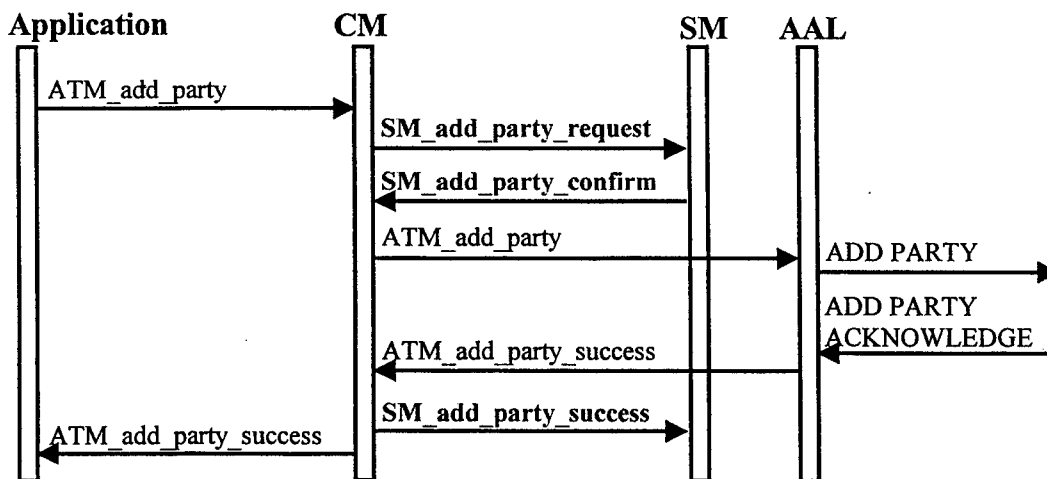


Figure 23. Adding a Party to Point-to-Multipoint Connection

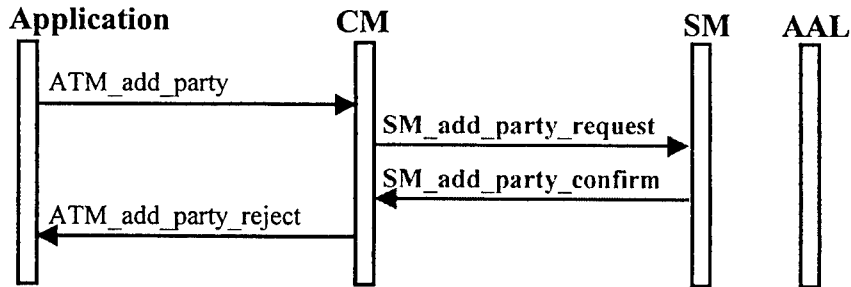


Figure 24. Security Rejection of Adding a Party

4.3.4 Leaf Initiated Join at the Root Node

The ATM Forum UNI 4.0 specifications provides for two types of leaf initiated join to point-to-multipoint connections: root leaf initiated join and network leaf initiated join [21]. Root leaf initiated join requires the participation of the root node of the point-to-multipoint connection in order to add a leaf node to the connection. Network leaf initiated joins are handled by the network, without the participation or knowledge of the root node. The root node can only control whether network leaf initiated joins are allowed for a particular connection. This information is specified in the initial SETUP request that is used to establish the connection. Once network leaf initiated joins are allowed for a connection, the root node has no control over or knowledge of the leaves that are added to the connection (other than those that are explicitly added by the root). Network leaf initiated joins are not appropriate in situations where the root node must maintain control over who can be added to the connection.

In Figure 25, a leaf adds itself to a point-to-multipoint connection using a root leaf initiated join.

1. Upon the arrival of the LEAF SETUP REQUEST indication from the network, the AAL passes the ATM_LIJ_join_requested indicator to the Connection Manager to notify it that a request to join a point-to-multipoint connection has been received. The Connection Manager passes the ATM_LIJ_join_requested indicator to the application.
2. The application proceeds to add the leaf node to the connection as described for Figure 21 above.

If the Security Manager rejects the addition of a leaf to a multiparty call, as shown in Figure 26, the connection attempt fails. In this case, the Connection Manager sends an ATM_LIJ_reject_leaf response to the network. The application would make the same response if it were to reject the leaf setup request.

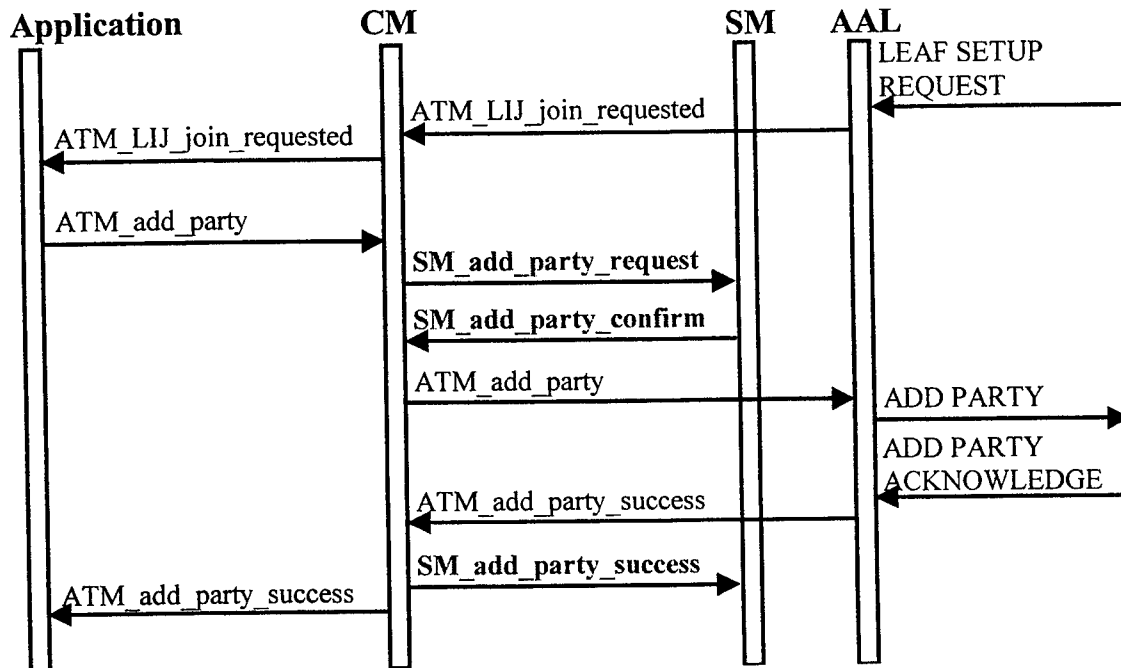


Figure 25. Leaf Initiated Join at the Root Node

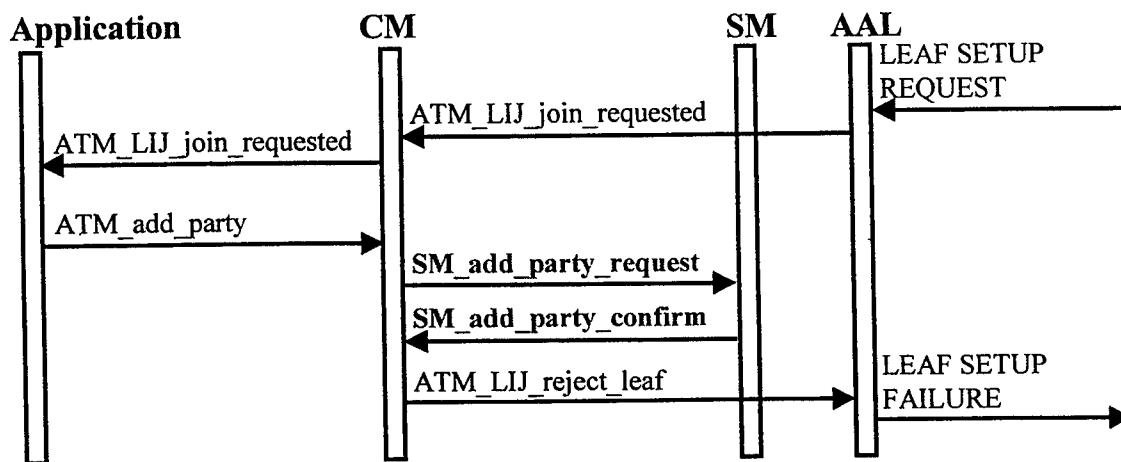


Figure 26. Security Rejection of a Leaf Initiated Join

4.3.5 Leaf Initiated Join at a Leaf Node

In Figure 27, a leaf node initiates a root leaf initiated join to a point-to-multipoint connection as follows:

1. The ATM_associate_endpoint request associates an API endpoint with the potential API connections that will use that endpoint. The application passes this request to the Connection Manager, and the Connection Manager passes it to the AAL.

2. If the application is security-aware, it can use the `ATM_set_security_attributes` and `ATM_query_security_attributes` requests to set and query security parameters for the connection.
3. The `ATM_prepare_incoming_call` request sets up the incoming call distribution tables. This request is passed first to the Connection Manager, and then by the Connection Manager to the AAL.
4. The application sends the `ATM_LIJ_request_join` request to the Connection Manager to attempt a leaf initiated join of a point-to-multipoint connection. The Connection Manager informs the Security Manager with a `SM_LIJ_request_join` message and passes the `ATM_LIJ_request_join` request on to the AAL, which sends out the appropriate LEAF SETUP REQUEST.
5. The AAL waits for the SETUP message corresponding to the previous LEAF SETUP REQUEST to arrive. The corresponding `ATM_arrival_incoming_call` indication is sent to the Connection Manager, which processes it like any other incoming call. See Section 4.3.2, steps 4 through 13, for details.

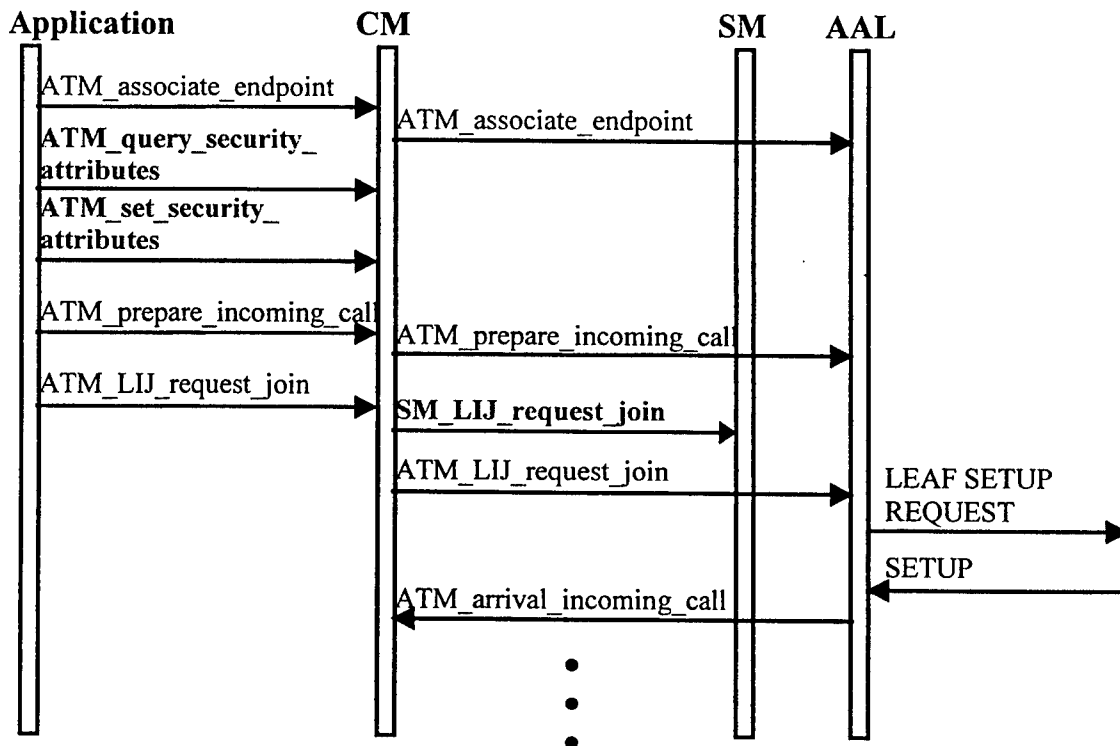


Figure 27. Leaf Initiated Join at a Leaf Node

If the Security Manager rejects the addition of the leaf to a multiparty call, as shown in Figure 28, the connection attempt fails. In this case, the Connection Manager sends an `ATM_LIJ_leaf_rejected` confirmation to the application and an

ATM_reject_incoming_call response to the network. The ATM_LIJ_leaf_rejected confirmation is the same confirmation that the application would receive if the network were to reject the addition of the leaf node. The ATM_reject_incoming_call response is the same response that the network would receive if the application were to reject the addition of the leaf.

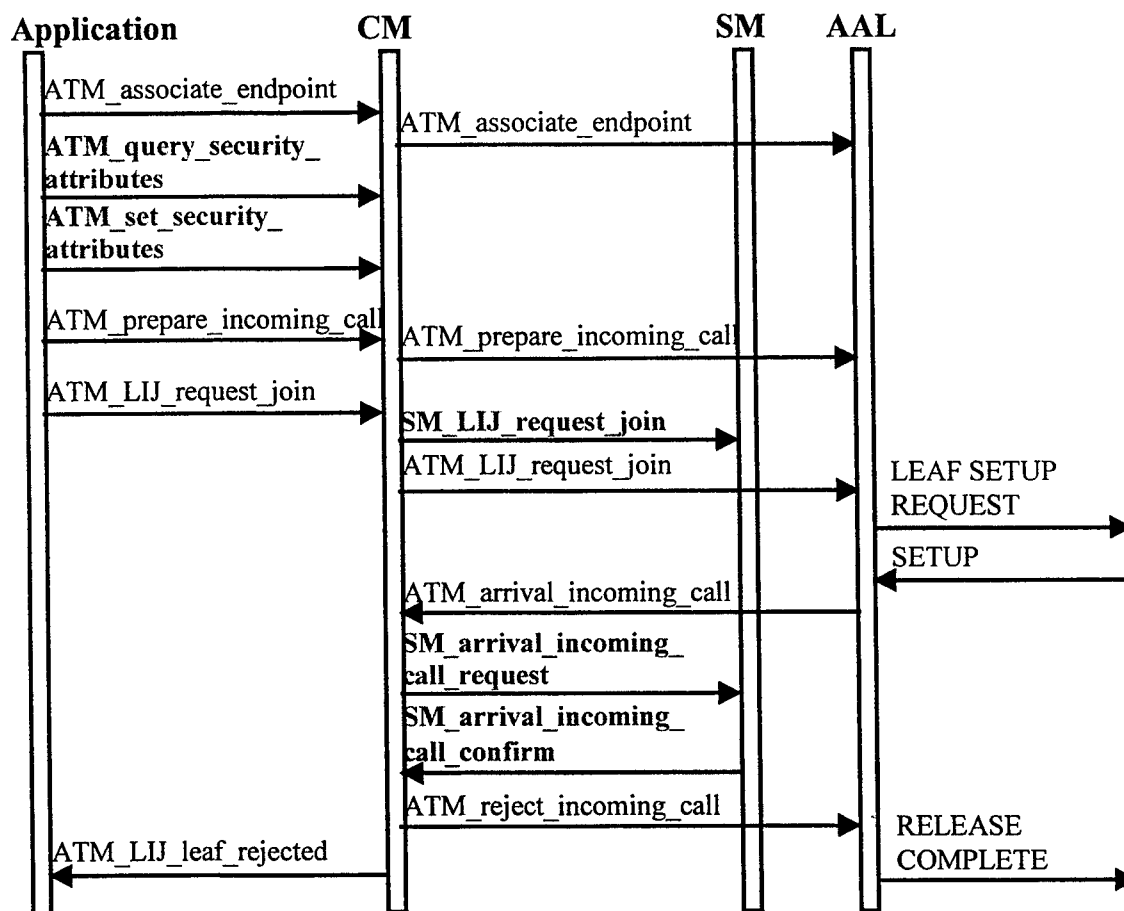


Figure 28. Security Rejection of Leaf Initiated Join at a Leaf Node

4.3.6 Sending and Receiving Data

In Figure 29, data is sent and received on an ATM connection as follows:

1. The application sends data on the established API connection using the ATM_send_data request. The application passes this request to the Connection Manager, and the Connection Manager passes it to the AAL. The AAL then passes the data to the network.
2. The AAL receives data from the network. The AAL sends the data to the Connection Manager using the ATM_receive_data indication and then the Connection Manager passes it to the application.

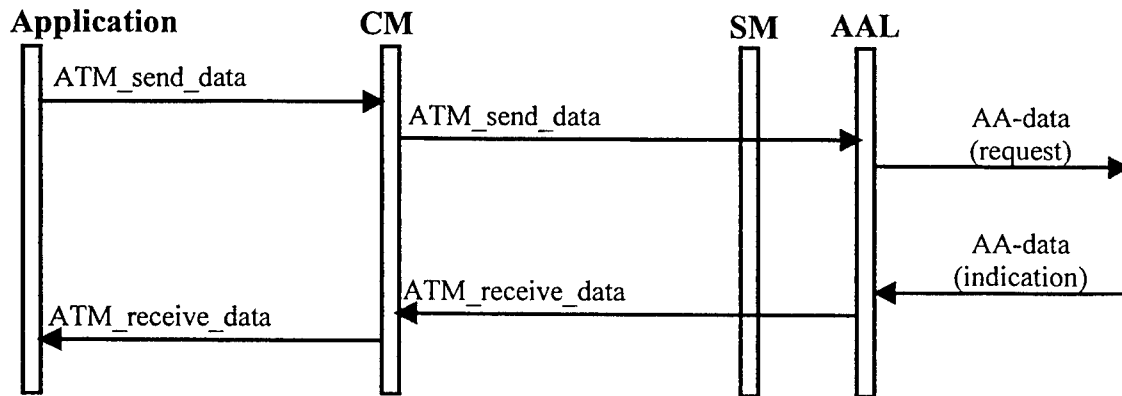


Figure 29. Sending and receiving data on a connection.

Note that the Connection Manager may apply security services to the protocol data units (PDUs) as they pass through it. For example, the Connection Manager may encrypt each outgoing PDU that it receives from the application and sends to the network, and may decrypt each incoming PDU it receives from the network and sends to the application.

4.3.7 Application Initiated Connection Release

In Figure 30, an application releases a connection as follows:

1. The application terminates an API connection by passing an `ATM_call_release` request to the Connection Manager.
2. The Connection Manager informs the Security Manager of the connection release using the `SM_call_release` message, and then passes the `ATM_call_release` request on to the AAL, which sends a release request to the network.

4.3.8 Network Initiated Connection Release

In Figure 31, the network releases a connection as follows:

1. The AAL receives a `RELEASE` indication from the network, passes the `ATM_call_release` indication to the Connection Manager, releases the connection, and passes the `RELEASE COMPLETE` confirmation to the network.
2. The Connection Manager informs the Security Manager of the connection release using the `SM_call_release` message, and then passes the `ATM_call_release` indication to the application.

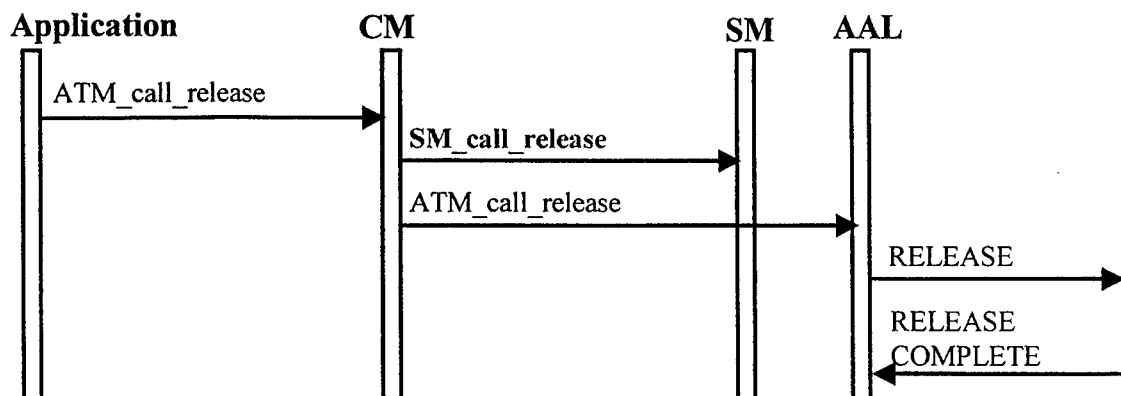


Figure 30. Connection Release by an Application.

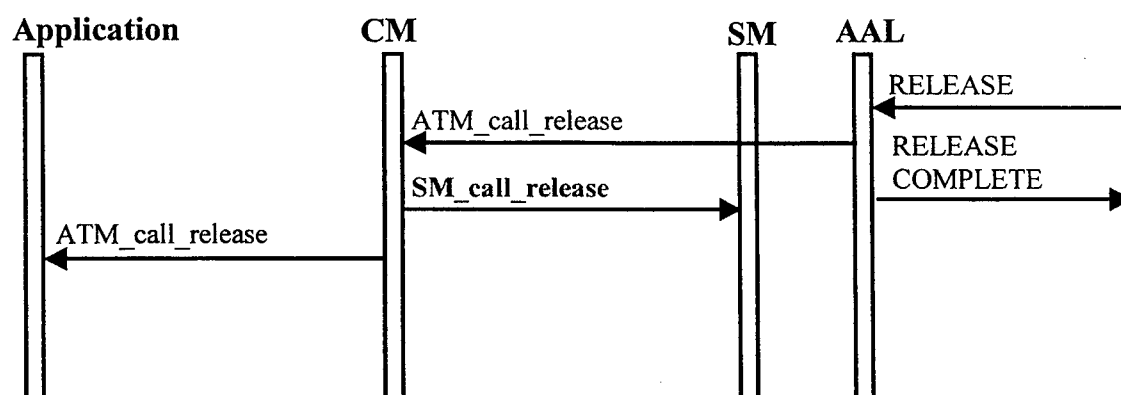


Figure 31. Connection Release from the network.

4.3.9 Dropping a Party from a Point-to-Multipoint Connection

In Figure 32, an application drops a party from a point-to-multipoint connection through the following sequence of steps:

1. The application passes the ATM_drop_party request, which drops a leaf from an existing point-to-multipoint connection, to the Connection Manager.
2. The Connection Manager interacts with the Security manager through the SM_drop_party_request and SM_drop_party_confirm messages. The Connection Manager uses the SM_drop_party_request to pass security-relevant information about the connection to the Security Manager. The Security Manager responds to the Connection Manager with a SM_drop_party_confirm message, which contains the approval or denial of the “drop party” request.
3. If the Security Manager approves the “drop party” request, the Connection Manager passes the ATM_drop_party request to the AAL. The AAL requests the dropping of the connection from the network, which in turn confirms the request.

4. Upon confirmation of the “drop party” success or denial from the network, the AAL informs the Connection Manager by passing back either the ATM_drop_party_success or the ATM_drop_party_denial confirmation. The Connection Manager in turn passes the confirmation back to the application, informing it of the success or failure of the “add party” attempt. The Connection Manager also informs the Security Manager of the success or failure of the “drop party” request using a SM_drop_party_success or SM_drop_party_failure message.

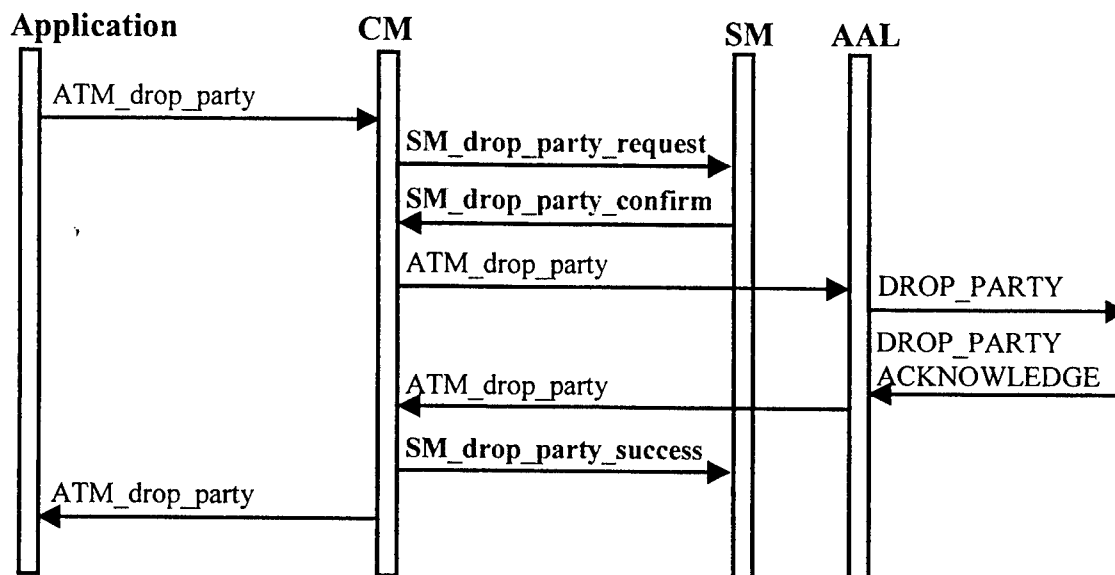


Figure 32. Dropping a Party from a Multipoint Connection

4.4 Security Management Workstation to Enclave Boundary Interface

In the ATM USA, enclave to network communication is handled on a per-connection basis. For each connection, both control and data channels must be established. The Security Manager, which approves the connection requested by the Connection Manager, in turn communicates with the gateway device, in order to open the control and data channels for the connection. The various calls associated with the opening and closing of these channels by the gateway device are listed in Table 13 below.

Table 13. Security Manager to Gateway Device Interface Primitives

ATM Security-Mediated Connection Establishment/Release Operation	Security Manager to Gateway Device Interface Primitives
Making or Accepting a Point-to-Point Call, or a Leaf Initiated Join at the Leaf Node	GW_open_control GW_open_data
Connection Release by an Application or the Network	GW_close_control GW_close_data

4.4.1 GW_open_control

In opening a control channel for a connection, the Security Manager passes the GW_open_control call to the enclave gateway device. The gateway device, upon receipt of this call, opens a control channel to the network for the connection.

4.4.2 GW_open_data

In opening a data channel for a connection, the Security Manager passes the GW_open_data call to the enclave gateway device. The gateway device, upon receipt of this call, opens a data channel to the network for the connection.

4.4.3 GW_close_control

In closing a control channel for a connection, the Security Manager passes the GW_close_control call to the enclave gateway device. The gateway device, upon receipt of this call, closes the connection's control channel to the network.

4.4.4 GW_close_data

In closing a data channel for a connection, the Security Manager passes the GW_close_data call to the enclave gateway device. The gateway device, upon receipt of this call, closes the connection's data channel to the network.

4.5 Security and Signaling Coordination with Gateway

In Figure 33 through Figure 40, the coordination of security and signaling calls between the application workstation, the Security Manager and the enclave gateway for various types of ATM connection establishment and release is specified. In Figure 19 through Figure 31, found in Section 4.3, the coordination of calls between the application, Connection Manager, Security Manager and ATM Adaptation Layer was given for the same types of ATM connection establishment and release. In those figures, the calls between the application, Connection Manager, and ATM Adaptation Layer – all of which reside internal to the application Workstation proper – were explicitly shown. The role of the enclave gateway in ATM connection establishment and release, however, was not shown.

In Figure 33 through Figure 40, the role of the enclave gateway in ATM connection establishment and release is made explicit. For the sake of clarity, the messaging between the application, Connection Manager, and ATM Adaptation Layer – all of which are internal to the application Workstation – is hidden. Only the calls to and from the Workstation proper are shown. Similarly, security interface calls are shown passing between the Security Manager and the enclave gateway in these diagrams, when in fact the Security Manager's Management Path Control component is responsible for communication with the gateway device.

4.5.1 Setup of Outgoing Connection

In Figure 33, the following sequence of steps is carried out to establish an outgoing connection. This sequence corresponds to steps 6-10 in Section 4.3.1, where Figure 19 corresponds to Figure 33.

1. In the process of setting up an outgoing connection, the Workstation passes security-relevant information about the connection to the Security Manager, using the `SM_outgoing_call_request` primitive. If the Security Manager approves the outgoing call request, it passes the `GW_open_control` primitive to the gateway, which opens a control channel for data channel negotiation. The Security Manager then passes the `SM_outgoing_call_confirm` primitive back to the Workstation, approving the connection request.
2. Upon acceptance of the `SM_outgoing_call_confirm` primitive from the Security Manager, the Workstation passes a `SETUP` connection setup request to the gateway, which forwards it to the network. The network responds to the gateway with a `CONNECT` confirmation of the connection setup request, which the gateway passes back to the Workstation. The Workstation then acknowledges this confirmation with a `CONNECT ACKNOWLEDGE` response to the gateway, which forwards it to the network.

3. The Workstation then passes the SM_call_active primitive to the Security Manager, informing it that the connection is in an active state. The Security Manager passes the GW_open_data primitive to the gateway, which then opens the data channel.

If the Security Manager rejects the call, as shown in Figure 34, it indicates this in the SM_outgoing_call_confirm primitive that it passes back to the Workstation, informing it that the connection attempt has failed.

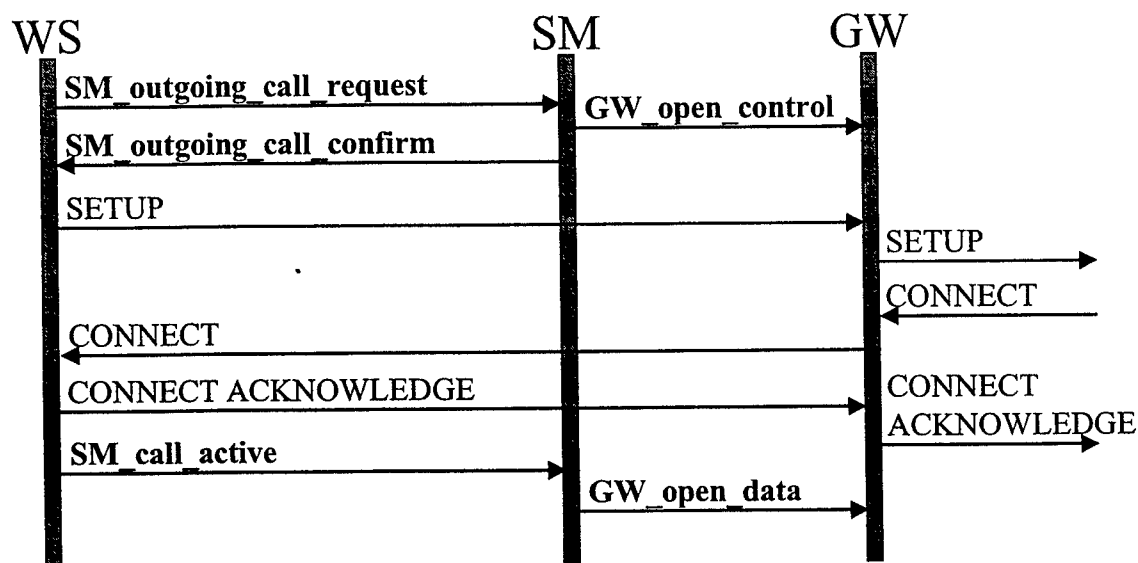


Figure 33. Outgoing Connection Setup

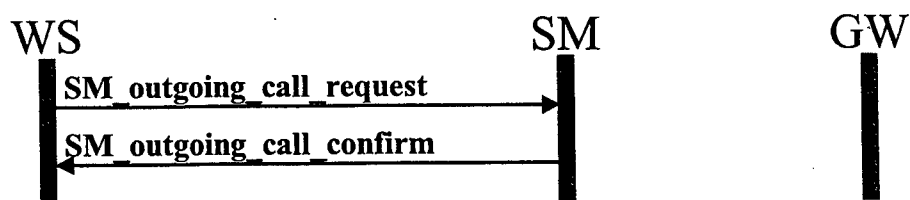


Figure 34. Outgoing Connection Setup Rejection

4.5.2 Setup of Incoming Connection

In Figure 35, the following sequence of steps is carried out to establish an incoming connection. This sequence corresponds to steps 6-10 in Section 4.3.2, where Figure 21 corresponds to Figure 35.

1. In preparation for setting up an incoming connection, the Workstation passes the SM_wait_on_incoming_call_request primitive to the Security Manager, informing the Security Manager of its anticipation of an incoming call. If the Security Manager approves the request, it passes the GW_open_control primitive to the gateway, which opens a control channel for data channel negotiation.

2. A SETUP connection setup indication arrives at the gateway from the network. The gateway forwards the indication to the Workstation.
3. After receiving the call notification from the gateway, the Workstation interacts with the Security Manager through the SM_arrival_incoming_call_request and SM_arrival_incoming_call_confirm messages. The Workstation uses the SM_arrival_incoming_call_request message to pass security-relevant information about the connection to the Security Manager. The Security Manager responds to the Workstation with a SM_arrival_incoming_call_confirm message containing the approval of the connection request.
4. The Workstation then interacts with the Security manager through the SM_accept_incoming_call_request and SM_accept_incoming_call_confirm messages. The Workstation uses the SM_accept_incoming_call_request message to pass any updated security-relevant information about the connection to the Security Manager. The Security Manager responds to the Workstation with a SM_accept_incoming_call_confirm message containing the approval of the connection request.
5. The Workstation then responds to the gateway with a CONNECT confirmation of the connection setup request, which the gateway passes on to the network. The network in turn passes a CONNECT ACKNOWLEDGE indication to the gateway, which forwards it to the Workstation.
6. The Workstation then passes the SM_call_active primitive to the Security Manager, informing it that the connection is in an active state. The Security Manager passes the GW_open_data primitive to the gateway, which then opens the data channel.

If the Security Manager rejects the incoming call in step 3, as shown in Figure 36, the connection attempt fails. In this case, the Workstation sends a RELEASE COMPLETE message to the gateway, which passes it to the network.

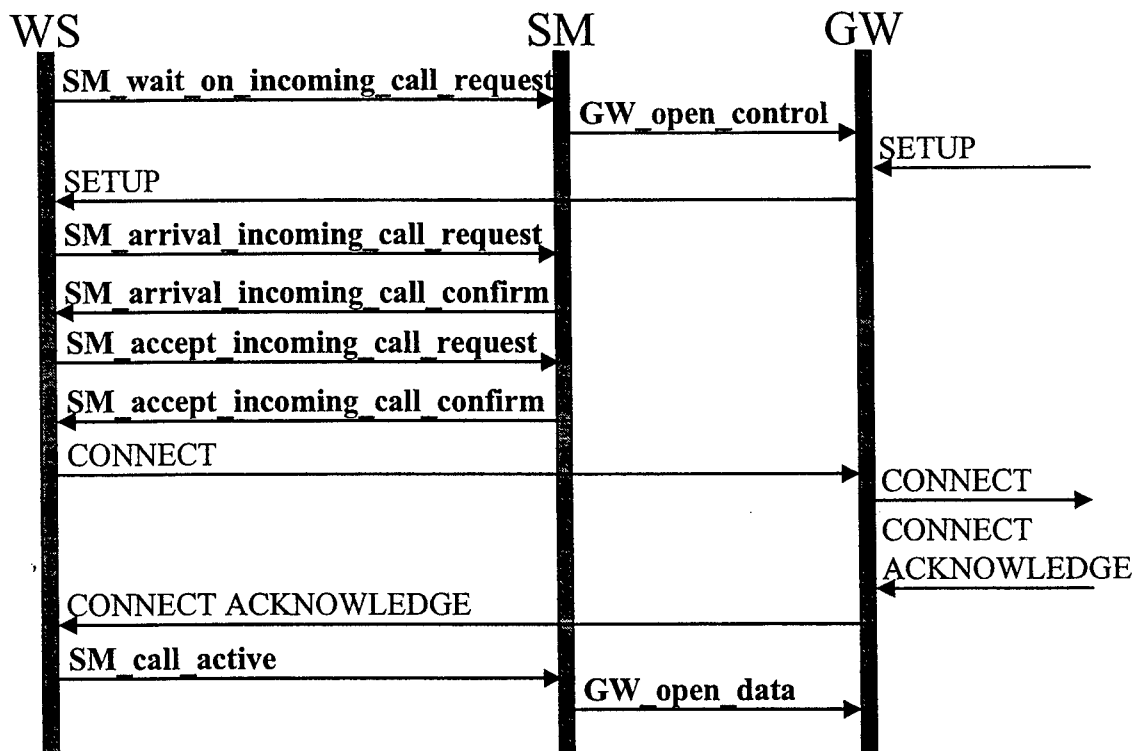


Figure 35. Incoming Connection Setup

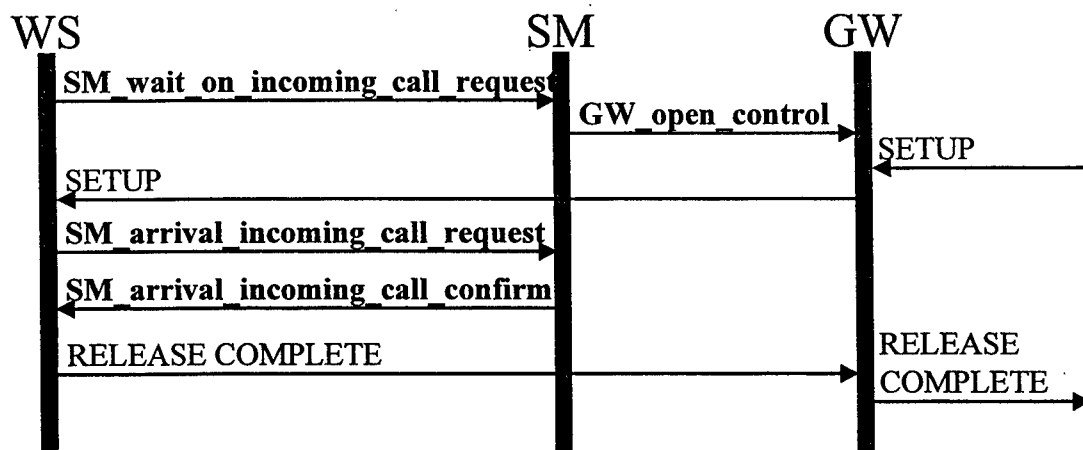


Figure 36. Incoming Connection Setup Rejection

4.5.3 Leaf Initiated Join at a Leaf Node

In Figure 37, the following sequence of steps is carried out to establish a leaf initiated join at a leaf node. This sequence corresponds to steps 1-2 in Section 4.3.5, where Figure 27 corresponds to Figure 37.

1. In preparation for setting up a leaf initiated join, the Workstation passes the SM_LIJ_request_join primitive to the Security Manager, informing the Security Manager of its request for a join. If the Security Manager approves the request, it passes the GW_open_control primitive to the gateway, which opens a control channel for data channel negotiation.
2. The Workstation then sends a LEAF SETUP REQUEST to the gateway, which passes it on to the network.
3. The subsequent steps are identical to steps 2-6, detailing an incoming connection setup, in Section 4.5.2.

If the Security Manager rejects the addition of the leaf to a multiparty call, as shown in Figure 38, the connection attempt fails, in an identical manner to that discussed in Section 4.5.2.

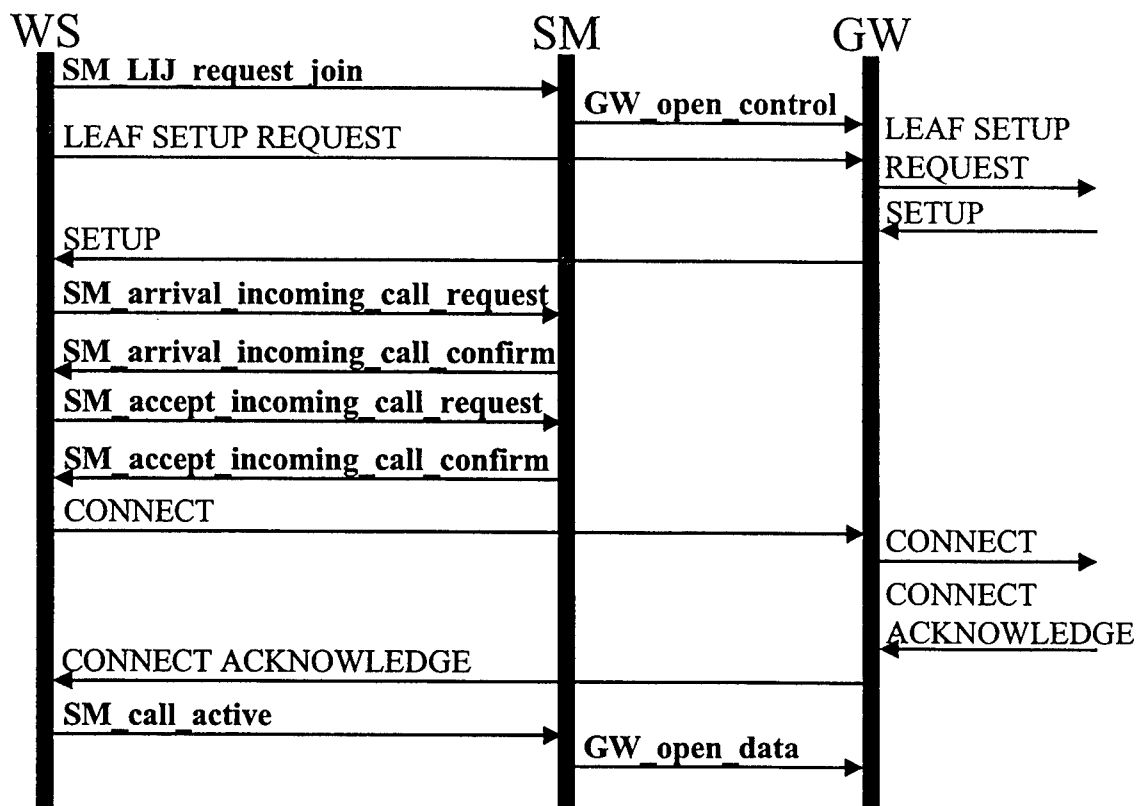


Figure 37. Leaf Initiated Join at a Leaf Node

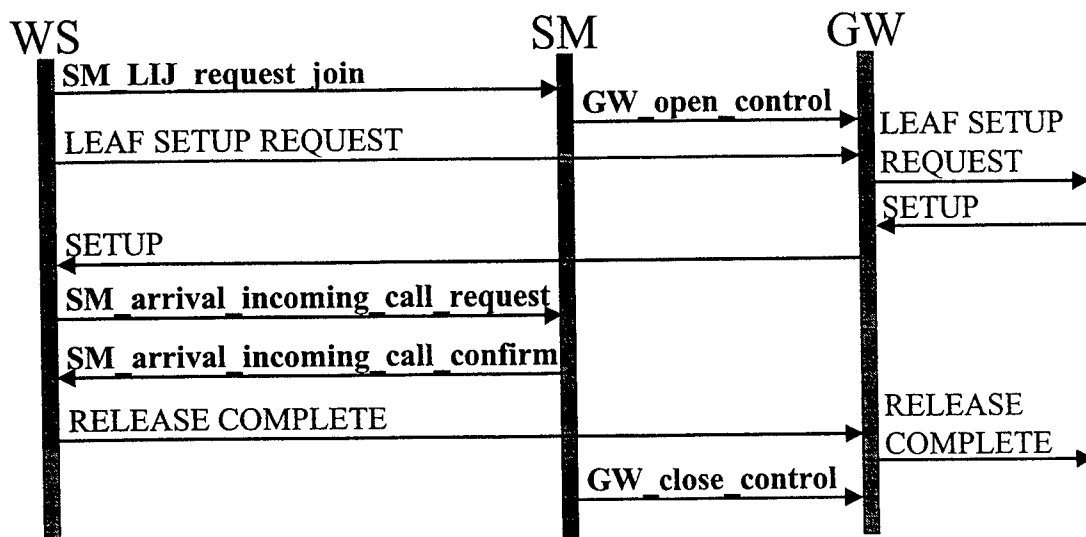


Figure 38. Leaf Initiated Join at a Leaf Node Rejection

4.5.4 Outgoing Release of Established Connection

In Figure 39, the following sequence of steps is carried out to release a connection from the Workstation. This sequence corresponds to steps 6-10 in Section 4.3.7, where Figure 30 corresponds to Figure 39.

1. In the process of releasing an established connection, the Workstation informs the Security Manager of the connection release using the SM_call_release primitive. The Workstation then passes the RELEASE request to the gateway, which forwards it to the network.
2. The Security Manager passes the GW_close_data primitive to the gateway after a fixed time interval has elapsed following the receipt of the SM_call_release request from the Workstation. The gateway, upon receiving the GW_close_data request, closes the data channel of the connection.
3. The network responds to the gateway with a RELEASE COMPLETE message, which the gateway passes on the Workstation.
4. After another fixed timing interval has elapsed, following the sending of the GW_close_data primitive, the Security Manager passes the GW_close_control primitive to the gateway. The gateway, upon receiving the request, closes the control channel of the connection.

4.5.5 Incoming Release of Established Connection

In Figure 40, the following sequence of steps is carried out to release a connection from the network. This sequence corresponds to steps 6-10 in Section 4.3.8, where Figure 31 corresponds to Figure 40.

1. In the process of releasing an established connection, the network passes the RELEASE indication to the gateway, which forwards it to the Workstation.
2. The Workstation informs the Security Manager of the connection release using the SM_call_release primitive. Upon receipt of this request, the Security Manager passes the GW_close_data primitive to the gateway, which then closes the data channel of the connection.
3. After a fixed timing interval, the Workstation responds to the network's RELEASE indication, sending a RELEASE COMPLETE response to the gateway, which then forwards it to the network.
4. After a fixed timing interval has elapsed, following the sending of the GW_close_data primitive, the Security Manager passes the GW_close_control primitive to the gateway. The gateway, upon receiving the request, closes the control channel of the connection.

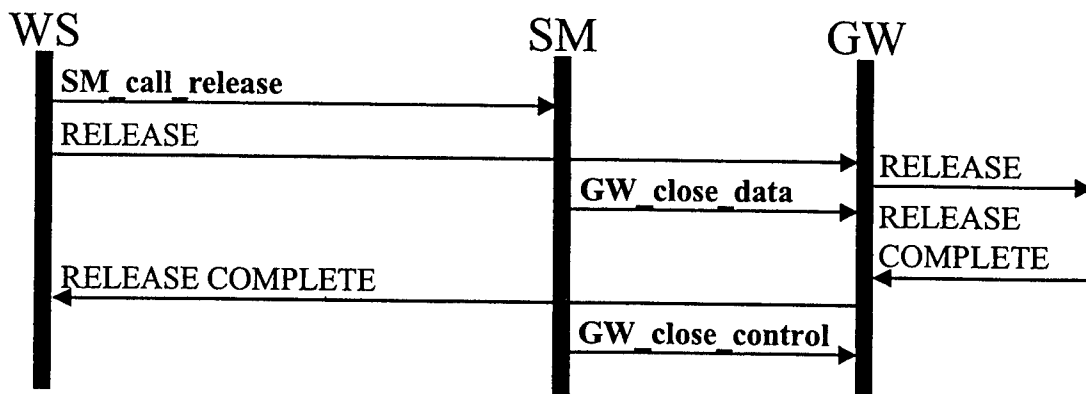


Figure 39. Outgoing Connection Release

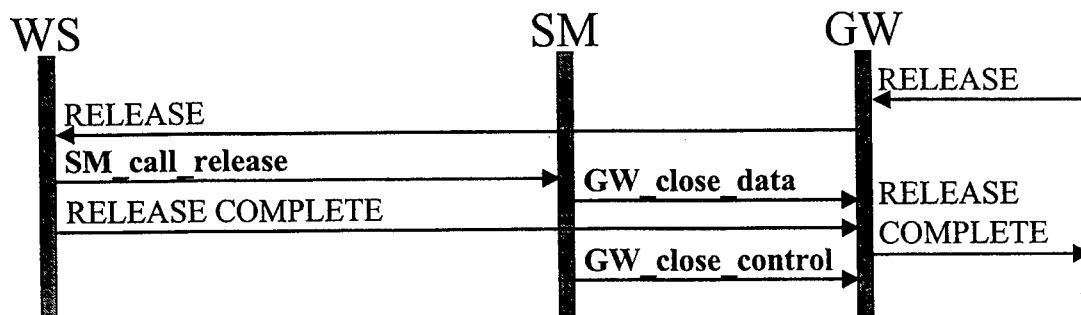


Figure 40. Incoming Connection Release

4.6 Security Management Workstation to Security Management Workstation Interface

The ATM USA might also support an interface that would enable communication between the security managers at different enclaves. Security managers could exchange information relating to network topology, security services, certificates, key management, and network management, as well as other possible topics. While not part of the current ATM USA architecture, such an interface would extend its capabilities, allowing it to provide more appropriate and flexible services for a given network.

4.7 Security Manager to FASTLANE Gateway Device Interface

In the network management scheme that supports SNMP, network managers manage and communicate with network agents. Both managers and agents implement a Management Information Base (MIB), a structured collection of managed objects, specified in the ASN.1 notation, which hold values representing managed resources. The Simple Network Management Protocol, SNMP, is a basic mechanism, supporting GET, SET and TRAP operations on MIB objects, for the exchange of management information between a manager and an agent.

As specified in the GTE "FASTLANE Training" course [22], FASTLANE (along with TACLANE, for everything that follows) is SNMP compatible. In short, an authorized SNMP-enabled network manager may perform a subset of the FASTLANE configuration and display functions. These functions, covered in the GTE FASTLANE User's Manual, are normally manipulated locally, through the FASTLANE HMIs (Human Machine Interfaces). A network manager could perform the following FASTLANE functions: Create/Delete Key Assignments, Change Device Modes, Display Datastores, and Configure Hardware/Software. A FASTLANE device implements a single SNMP agent, remotely accessible from both its red and black sides. An SNMP network manager may interact with the FASTLANE SNMP agent to perform the functions mentioned above.

In the ATM USA, an enclave's Security Manager interacts with the enclave gateway device in the course of establishing a secure ATM connection. This interaction, of course, must be through a pre-established protocol supported by both the Security

Manager and the gateway device. If the gateway device is a FASTLANE ATM cell encryptor, SNMP is the natural management protocol to use, since FASTLANE already supports it. Given this, the Security Manager would then implement a SNMP network manager to carry out remote management functions on the FASTLANE device. Of the possible functions mentioned above, the two that matter for the establishment of ATM connections are Create/Delete PVCs and Activate/Deactivate PVCs, outlined in further detail in the GTE FASTLANE User's Manual [23].

There are two possible limitations to such a scheme. First, remote management of a FASTLANE device requires authentication and access control of the SNMP manager, which in practice implies placing a FASTLANE device in front of an SNMP manager to provide security services. Within an enclave, such security may be unnecessary, but it is not clear that the current configuration of the SNMP-enabled FASTLANE device will allow its circumvention. Second, in addition to the management of PVCs, it is also necessary – for the general establishment of ATM connections – that it be possible to remotely manage SVCs as well. Although the GTE FASTLANE User's Manual does not specify explicit functions for managing SVCs (perhaps not too surprisingly, since SVC establishment is largely a Control Plane issue), it does mention SVCs in the management of CIKs (Cryptographic Ignition Keys), where either PVCs or SVCs may be assigned to filled preplaced keys, in order to enable the establishment of ATM connections. This strongly suggests that FASTLANE supports SVCs. Additional evidence is provided by the GTE "FASTLANE Training" course, where it is explicitly stated that both PVCs and SVCs are supported in the interaction of an SNMP manager, fronted by a FASTLANE device, with a remotely managed FASTLANE device.

In closing, it appears quite feasible to use SNMP to remotely manage ATM connection establishment for a FASTLANE gateway device by an enclave Security Manager.

4.8 An Overview of ATM PNNI Standard

The user-network interface (UNI) protocols [24, 25] enable an end system to establish connections with other end systems through the ATM network. Similarly, the ATM private network-network interface (PNNI) protocols are used to route ATM signaling requests between switching systems. A switching system can be a single switch or a network of multiple switches, all of which are under a single control system. The ATM Forum PNNI 1.0 specifications [11] define two categories of protocols for use between switching systems. The PNNI routing protocol is used to distribute topology information between switching systems. The PNNI signaling protocol establishes point-to-point (point-to-multipoint) connections across an ATM network.

PNNI Version 1.0 [11] has been designed to support all UNI 3.1 capabilities and some UNI 4.0 capabilities. It is scalable to very large networks, and supports QoS-based routing, where QoS based routing is a routing algorithm that ensures that routes, which meet the QoS requirements, are selected for each connection. PNNI's dynamic routing protocol enables it to route around failed links and links with insufficient resources. The key to the scalability of the PNNI is the hierarchical organization of the network, with a

summarization of reachability information between levels in the hierarchy. At each level in the hierarchy, PNNI defines a uniform network model that explains how the level operates and how the nodes at that level interact with those below and above its level.

PNNI views the network as a collection of peer groups. At the lowest level, peer groups are formed as collections of switches. All nodes in a peer group have complete state information on each other, and a peer group ID identifies each peer group. Peer groups are in turn organized hierarchically into higher levels of peer groups, which are parent peer groups of the lower-level peer groups. Within its parent peer group, each peer group is represented by a single logical entity called the logical group node (LGN), which acts as a normal node in the parent peer group. The major PNNI routing functions are following:

1. **Peer Group Leader Election:** Each peer group uses a well-defined algorithm to elect its peer group leader (PGL). Sometimes preference for the PGL is established through configuration. The PGL for a peer group represents that peer group at the next level of the hierarchy. In addition, the PGL is responsible for executing the functions of the LGN in the next hierarchical level, such as aggregation and distribution of information for maintaining the PNNI hierarchy.
2. **Hello Protocol:** Logical group nodes exchange hello packets to discover and verify the identity of their neighbors. It is through the hello message that the version of the PNNI that an LGN supports is identified. In addition, through the hello message a pair of neighboring LGNs can discover whether they are in the same peer group or not. An LGN that is adjacent to another LGN that is in another peer group is called a border node. Logical links between nodes in the same peer group are called horizontal links or inside links, and logical links between nodes in different peer groups are called outside links.
3. **Topology State Database Exchange:** This mechanism enables the nodes in a peer group to synchronize their topology databases. The topology database includes topology state information (a node's characteristics and link state parameters) and reachability information (addresses and address prefixes that describe destinations to which calls can be routed). This information is bundled in PNNI topology state elements that are flooded throughout the peer group.
4. **Topology State Summarization:** Topology state summarization permits the amount of information describing a peer group to be reduced as the information is passed up the routing hierarchy. The aggregation process is as follows: A child peer group is represented by a logical group node in the parent peer group through the process of nodal aggregation. Similarly, a set of links between two peer groups can be represented by a single logical link through the process of link aggregation. Finally, ATM addresses that are reachable from a peer group can be represented by a single address prefix. In this way, higher level entities in the hierarchy see only a summary of the state of the lower level entities.

5. Hierarchical Path Determination: PNNI uses source routing for connection setup. The node through which a peer group is entered (the ingress node of the peer group) is responsible for selecting the entire path across that peer group. The path is encoded as a designated transit list (DTL) that specifies each node used in transit across the peer group. If a node along the path is unable to follow the DTL for a connection setup request due to lack of resources, the node refuses the request and must crank back the request to the node that created the DTL.

Signaling protocol is used for call/connection establishment and clearing. PNNI signaling uses a subset of the UNI 4.0 signaling. It uses information gathered by PNNI routing; specifically, it uses route calculations derived from reachability, and resource information that is dynamically maintained by PNNI routing. A hierarchically complete source route is expressed as a sequence of DTLs ordered from lowest to highest peer group level and is organized as a stack with the DTL at the top of the stack corresponding to the lowest level peer group.

When a call arrives at a node that does not have the resources to service the call, the switch cranks the call back to the node that created the DTL. Crank back is a mechanism that allows a connection that is block along a selected path to be rolled back to a node earlier in the path, and eventually to the node that created the DTL. From that node, another path to the destination can be chosen which avoids the block node or link.

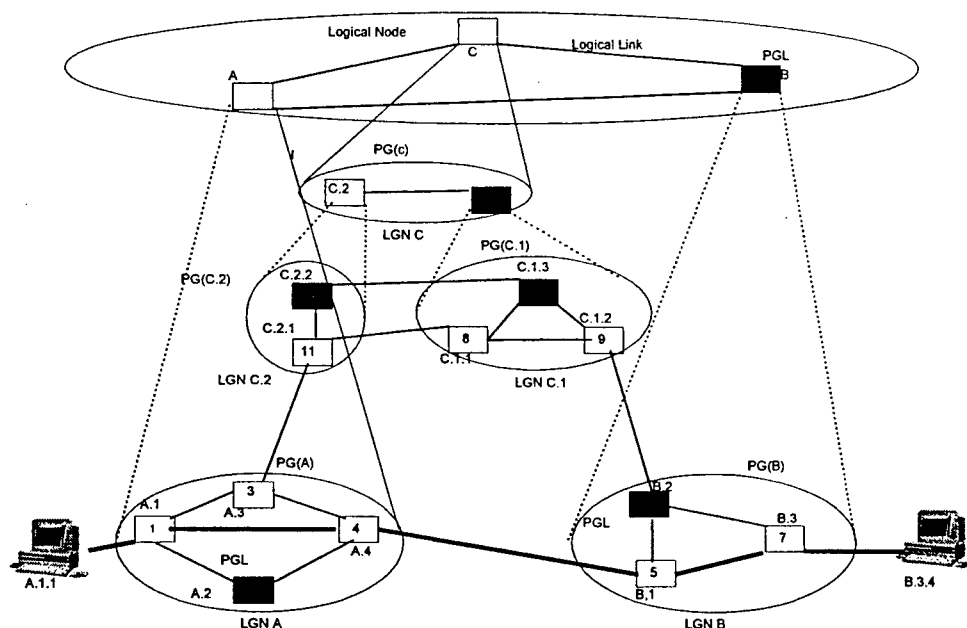


Figure 41. PNNI Network Hierarchical Abstraction

Figure 41 shows the network abstraction at different levels in the PNNI hierarchy. The figure shows how PG (C.1) and PG (C.2) have been abstracted to form the peer group PG (C) (i.e., LGN C) with the logical node C.1 as the peer group leader. The three logical nodes A, B, and C form the highest level peer group with logical group node B as the

peer group leader. Note that PGL functions for this level are implemented in node 6 (B.2), since it executes the LGN functions and the PGL functions for its peer group. Similarly, the PGL functions for LGN C are implemented in node 10 (C.1.3).

As an example of a connection establishment, consider an end system A.1.1, attached to switch A.1, which wants to establish a connection with end system B.3.4, attached to switch B.3. On receiving a UNI SETUP message from A.1.1 to A.1, A.1 constructs the necessary DTLs. There are four possible paths to B.3; (A.1, A.3, C, B), (A.1, A.3, A.4, B), (A.1, A.4, B), and (A.1, A.2, A.4, B). Assume that based on cost and policy, the path (A.1, A.4, B) is chosen. The SETUP message is successfully received by B.3 through A.4. The details of this scenario are as follows: On receiving the message, A.4 observes that the next address is B. Since it is adjacent to B through B.1, it forwards the message to B.1 that forwards it to B.3. Here, B.1 is responsible for constructing the detailed DTL for the LGN B. Usually, more than one DTL is defined, one for each level of the hierarchy. Each DTL, which specifies a route within one peer group, is organized as a stack that is a current transit pointer that indicates which element in the list is currently being visited at that level.

5 Conclusion

ATM USA provides the framework for building security solutions for DoD ATM users. Enclave security policies manage access to ATM services and the application of appropriate security services for these services. This architecture can be configured to support encryption devices at the local workstation, at the ATM switch, and at enclave boundaries (including cell and link encryptors). The enclave security policy can be used to provide a fine grain control over who has access to ATM services, the type and nature of the services that can be accessed, when they can be accessed, etc. The policy can also provide flexibility to balance the quality of service with the quality of protection for a connection and can provide dynamic management of both quality of service and quality of protection. The ATM USA is also compatible with emerging ATM standards, providing a solid basis for future compatibility as these standards develop in the future.

In addition, we have specified the ATM USA security extension to the ATM Native Services API. In addition to specifying the two API primitives, we defined how these primitives are invoked by an application in conjunction with its Connection Manager to manage security services. We also defined how the Connection Manager and the Security Manager interact with the application and network entities in the coordination of security with various types of ATM signaling activities. Further, we show the relationship between WinSock2 calls used by a Windows application and the underlying ATM Native Services primitives. In conjunction with this, we define the WinSock2 call security extensions. Finally, we define the security parameters, which form the options for the both the WinSock2 and the ATM API security extension, along with their possible values.

6 Glossary

AAL – ATM Adaptation Layer. A set of internationally standardized protocols and formats that define support for circuit emulation, packet video and audio, and connection-oriented and connectionless data services.

API – Application Programming Interface. An interface allowing an application to gain access to well-defined ATM-based capabilities.

ATM – Asynchronous Transfer Mode. A high-speed connection-oriented multiplexing and switching method specified in international standards utilizing fixed length cells to support multiple types of traffic. It is asynchronous in the sense that cells carrying user data need not be periodic.

CBC – Cipher Block Chaining. A mode of operation for block ciphers (e.g., DES and FEAL).

DES – Data Encryption Standard. A US standard (published by NIST) for data encryption.

DES40 – DES with a forty-bit effective key.

Diffie-Hellman – The first public key algorithm. Used primarily for symmetric key exchange.

DSA – Digital Signature Algorithm. The algorithm specified by the Digital Signature Standard, a US standard (published by NIST) for digital signatures.

ECB – Electronic Code Book. A mode of operation for block ciphers (e.g., DES and FEAL).

Elliptic Curve – Elliptic Curve Cryptosystem. A public-key cryptosystem.

ESIGN – Efficient digital SIGNature scheme. A digital signature algorithm.

FEAL – Fast data Encipherment ALgorithm. An encryption algorithm.

Firewalls, Guards, & Encryptors – Different types of gateway devices that may appear in the ATM USA enclave architecture.

HMAC – A Hash-based Message Authentication Code (see MAC), used in combination with a specific hash algorithm.

Identity-based Access Control – A method of access control involving rules based upon the identities of users and objects.

Label-based Access Control – A method of access control involving rules based upon levels and categories labeling objects and assigned to users.

MAC – Message Authentication Code. A mechanism for providing message integrity and authenticity.

MD5 – Message Digest 5. A hash algorithm that is typically used when generating digital signatures.

MIB – Management Information Base. A network topology database, used in the context of SNMP, the Simple Network Management Protocol.

NNI – Network-to-Network Interface. An interface allowing communication across distinct networks. Both public and private NNIs are used.

PVC – Permanent Virtual Circuit. A logical dedicated circuit between two user ports in a point-to-point configuration.

QoP – Quality of Protection. A measurement of desired protection for an ATM connection.

QoS – Quality of Service. A measurement of desired service for an ATM connection.

RIPEMD – A one-way hash algorithm.

Role-based Access Control – A method of access control involving rules based upon roles and properties assigned to users and objects.

RSA – An encryption/digital signature algorithm invented by Rivest, Shamir, and Adleman.

Security Management Workstation – The Security Management Workstation enforces the enclave security policy and governs the application of security services necessary for policy enforcement.

SHA-1 – Secure Hash Algorithm (Revision 1). The hash algorithm specified by the Digital Signature Standard.

SKE – A session key update algorithm, used in conjunction with a secure hash algorithm for updating data confidentiality and data integrity session keys.

SVC – Switched Virtual Circuit. Virtual circuits similar to PVCs, but established on a call-by-call basis.

Triple-DES – An algorithm, using DES, involving triple encryption for additional security.

User Application – In the ATM USA, a user application resides on an enclave user workstation. User applications engaged in ATM-based communication are subject to ATM USA security controls upon ATM connections, and may be provided ATM communication security services.

User Workstation – In the ATM USA, a user workstation encompasses user applications, local security services, and a Connection Manager, which manages ATM connections for applications residing on the workstation.

VCI – Virtual Channel Identifier. In ATM, a field within the cell header that is used to switch virtual channels.

VPI – Virtual Path Identifier. In ATM, a field within the cell header that is used to switch virtual paths, defined as groups of virtual channels.

X.509 Certificate – An RSA standard public-key certificate, used for establishing the trusted binding of a public key to an individual.

7 References

1. A. Danthine, Y. Baguette, *et al.* *The OSI95 Connection-Mode Transport Service - The Enhanced QoS.* in *The Fourth IFIP Conference on High Performance Networking*. 1992.
2. J. Jung and D. Seret. *Quality of Service in B-ISDN and Relation with Network Management.* in *Proc. IEEE ICC*. 1992. Chicago.
3. K. Nahrstedt and J. Smith, *The QoS Broker*. IEEE Multimedia, 1995(Spring).
4. D. Anderson, *Metascheduling for Continuous Media*. ACM Transaction on Computer Systems, 1993(August).
5. W. Tawbi, L. Fedaoui, and E. Horlait. *Dynamic QoS Issues in Distributed Multimedia Systems.* in *Second International Conference on Broadband Islands*. 1993. Athens.
6. A. Hafid, G. Bochmann, and B. Kerherve. *A Quality of Service Negotiation Procedure for Distributed Multimedia Presentation Application.* in *Proceedings of HPDC-5*. 1996. Syracuse.
7. O. Gerstel, I. Cidon, and S. Zaks, *The Layout of Virtual Paths in ATM Networks*. IEEE/ACM Transactions on Networking, 1996(December).
8. K. Murakami, *Virtual Path Routing for Survivable ATM Networks*. IEEE/ACM Transactions on Networking, 1996. 4(2): p. 22-38.
9. A. Ahuja, S. Keshav R, and H. Saran, *Design, Implementation, and Performance Measurement of a Native-Mode ATM Transport Layer (Extended Version)*. IEEE/ACM Transactions on Networking, 1996. 4(4): p. 502-515.
10. L. Zhang and S. Deering *et al.*, *RSVP: A New Resource Reservation Protocol*. IEEE Network Magazine, 1994(April).
11. ATM Forum, "ATM P-NNI Specification, Phase 1.0," March, 1996.
12. ATM Forum, "ATM Security Specification, Version 1.0," Report STR-SECURITY-01.00, December, 1997.
13. ATM Forum, "Native ATM Services: Semantic Description Version 1.0," Report af-saa-0048.000, February, 1996.

14. ATM Forum, "Mapping of the ATM Forum SAA/API Semantic Description to the WinSock2 API," Report atm96-0191, February, 1996.
15. X/Open, "XNET ATM Extensions, Appendix X: Use of XTI to Access ATM," Report atm96-1169 Draft, February, 1996.
16. WinSock Group, "Windows Sockets 2 Application Programming Interface, Revision 2.2.1," May 2, 1997.
17. WinSock Group, "Windows Sockets 2 Service Provider Interface, Revision 2.2.1," May 2, 1997.
18. WinSock Group, "Windows Sockets2 Protocol Specific Annex, Revision 2.0.3," May 10, 1996.
19. Stardust Technologies, "Stardust WinSock Component Architecture: Plug-In Developer Guide," Document 082296-03, 1996.
20. ATM Forum, "ATM Security Framework, Version 1.0," Report STR-SEC-FRWK-01.00 Straw Vote, December, 1997.
21. ATM Forum Technical Committee, "ATM User-Network Interface Signalling Specification, Version 4.0," Report af-sig-0061.000, July, 1996.
22. G. Hamilton, *Fastlane Training course*, . 1997, GTE Corporation.
23. GTE Corporation, "FASTLANE User's Manual," GTE Corporation December 16, 1996.
24. ATM Forum, *ATM User-Network Interface Specification, Version 3.1*. 1995: Prentice Hall.
25. ATM Forum, "ATM User-Network Interface Specification, Version 3.1," August, 1994.
26. K. Dunn, "Global Grid Security Architecture," National Security Agency November 1, 1995.
27. D. Morris, "Tactical Security Product Concept for Global Grid," MITRE Corporation March 31, 1994.
28. P. Woodie, "MISSI Overview: Security for the Defense Information Infrastructure," National Security Agency January 31, 1996.

29. D. Morris, "Secure Tactical ATM Broadcast Concept," MITRE Corporation March 31, 1994.
30. Air Force, "Theater Battle Management Security Policy," (year of publication unknown).
31. N.J. Meier, N.M. Lorenz, and S.S. Spencer, "Global Grid: The New InfoSec Challenge (draft)," National Security Agency
32. MIL-STD-187-700 Working Group, "Interoperability and Performance Standards for the Defense Information System," January 10, 1994.
33. S. Chuang, "Securing ATM Networks," October 18, 1995.
34. A. Alles, "ATM Internetworking," Cisco Systems, Inc. May, 1995.
35. J. Hughes, "A High Speed Firewall Architecture for ATM/OC-3c," Network Systems Corporation February 9, 1996.
36. P.W. Dowd, J.T. McHenry, *et al.*, "A Method for High Performance Network Security for IP and non-IP Traffic in an ATM Environment," National Security Agency Report TR-R53-06-97, April 11, 1997.
37. R.J. Vetter, *ATM Concepts, Architectures and Protocols*. Communications of the ACM, 1995. **38**(2): p. 30-38.
38. B.G. Kim and P. Wang, *ATM Network: Goals and Challenges*. Communications of the ACM, 1995. **38**(2): p. 39-44.
39. D. Stevenson, N. Hillery, and G. Byrd, *Secure Communications in ATM Networks*. Communications of the ACM, 1995. **38**(2): p. 45-52.
40. S. Fotedar et al., *ATM Virtual Private Networks*. Communications of the ACM, 1995. **38**(2): p. 101-109.

8 Appendix - ATM Requirements Collection

This section specifies the set of requirements collected for Task 1 of the ATM project. The design of the ATM security services needs to take into account the security requirements of the military community. In particular, we have studied the current needs of the following standards:

- Defense Information System Network Security Architecture (DISN)
- Multilevel Information System Security Initiative (MISSI)
- Security Policy for Theater Battle Management C4I Architecture for Deployable Operations
- Global Grid Security Architecture

In addition, we include the traceability matrix relating the requirements of these programs to the ATM USA derived requirements given in Section 2.2.

8.1 Defense Information System Network Security Architecture (DISN)

Table 14 identifies the applicable requirements extracted from the Interoperability and Performance Standards for the Defense Information System. Page and section numbers can be traced to the original document [32] and a reference field indicating any related documents.

Table 14. DISN Requirements

Req. #	Description	Source	Pg.	Section	Reference
I-1	Security - Protection measures are required (a) to prevent unauthorized access to the system and ensure the confidentiality of the information it carries, and (b) to preserve the integrity of the data and mitigate against denial of service	MIL-187-700A	62	4.5.3.5	
I-2	Information Security - The design of information systems shall allow the incorporation of communications security (COMSEC) and computer security (COMPUSEC) to protect information against unauthorized disclosure, transfer, modification or destruction.	MIL-187-700A	49	4.3.1	

Req. #	Description	Source	Pg.	Section	Reference
I-3	Communication Security - Provisions for COMSEC shall include crypto security, transmission security (TRANSEC), emission security (EMSEC), and physical security.	MIL-187-700A	49	4.3.1.1	
I-4	Crypto Security - Information systems shall provide internal or external crypto equipment. Digital interfaces to external crypto equipment shall be in accordance with MIL-STD-188-114.	MIL-187-700A	50	4.3.1.1.1	MIL-STD-188-114.
I-5	Transmission Security - HF radio anti-jam (AJ) systems shall comply with the TRANSEC algorithm provisions in MIL-STD-188-148. VHF radios shall comply with (JIEO) specification 9001. UHF radios shall comply with STANAG 4372. Standards for SATCOM AJ systems shall be based on existing UHF, SHF, and EHF common-user DOD satellite systems.	MIL-187-700A	50	4.3.1.1.2	MIL-STD-188-148, JIEO-9001, STANAG 4372
I-6	Emission Security - Compromising emanations shall be controlled within applicable TEMPEST criteria in the current edition of NSTISSAM TEMPEST/1-91.	MIL-187-700A	50	4.3.1.1.3	NSTISSAM TEMPEST 1-91
I-7	Physical Security - Systems shall have appropriate tamper-resistant design features and tamper-detection mechanisms.	MIL-187-700A	50	4.3.1.1.4	
I-8	Computer Security - Computer systems shall comply with applicable provisions of DOD 5200.28-STD.	MIL-187-700A	50	4.3.1.2	DOD 5200.28-STD
I-9	Voice - For end-to-end encrypted voice calls, and for non-secure voice calls between deployed and fixed subscribers continuous bit rate (CBR) shall be used.	MIL-187-700A	102	5.6.3.1.1	

Req. #	Description	Source	Pg.	Section	Reference
I-10	End-to-End encrypted voice service - Service between tactical and strategic networks shall be available if the subscriber terminals contain a common voice-encoding algorithm and a common crypto algorithm, and are capable of mode negotiation. End-to-end encrypted voice calls shall be treated as a data service. BCI shall be preserved to maintain cryptographic synchronization between the calling and called secure-voice terminals.	MIL-187-700A	43	4.1.6.2	
I-11	End-to-end encrypted voice service in ATM networks - Shall use AAL Type 1 CBR service for access to and exit from ATM networks, to maintain BCI. The AAL Type 1 protocol may be implemented in either the telephone terminal or a terminal adapter. The terminal adapter may be located in the ATM switch.	MIL-187-700A	43	4.1.2.2.3	
I-12	Security Management - The network security manager shall be able to grant or restrict access to the entire network or selected critical parts of the network, such as the NM information base.	MIL-187-700A	113	5.7.2.3 e	
I-13	NM Security - Security of the NM systems and information, and management of the security mechanisms that protect user traffic, shall be in accordance with MIL-HDBK-2045-1351, the section titled NM Security.	MIL-187-700A	113	5.7.2.4	with MIL-HDBK-2045-1351

8.2 Multilevel Information System Security Initiative (MISSI)

Table 15 identifies the applicable requirements extracted from MISSI Fortezza for Classified (FFC) System Description. Page and section numbers can be traced to the original document [28], as well as a reference field that specifies related documents.

Table 15. MISSI Requirements

Req. #	Description	Source	Pg.	Section	Reference
M-1	General - The FFC system shall provide positive Identification and Authentication (I&A) for messages traversing the enclave boundary in accordance with local security policy.	FFC System Description Rev 2 May 6, 1996	7	5.1.1	
M-2	General - The FFC system shall re-grade messages in accordance with local security policy as follows: the Guard shall downgrade messages destined for lower level enclaves, and it shall upgrade messages entering from other lower level enclaves.	FFC System Description Rev 2 May 6, 1996	7	5.1.1	
M-3	General - The FFC system shall admit or reject unprotected unclassified messages in accordance with enclave security policy	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	7	5.1.1	
M-4	General - The FFC system shall provide writer to reader security for services applied to the message content.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.1	
M-5	I&A - The FFC system shall require local (physical) user authentication for access to the secure messaging application.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.2	
M-6	I&A - The FFC system component will require that a user or role be uniquely identified when performing functions on behalf of the user or role (i.e., all actions must be associated with a unique user),	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.2	

Req. #	Description	Source	Pg.	Section	Reference
M-7	I&A - The FFC system component requires the ability to identify and authenticate the source of information passed between FFC system components.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.2	
M-8	I&A - The FFC system shall provide Data Origin Authentication	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.2	
M-9	Access Control - The FFC system shall provide Access Control for electronic messages.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.3	
M-10	Access Control - The FFC system shall provide the capability to restrict access to messages containing Compartmented information based on strong I&A and access control lists.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.3	
M-11	Non-Repudiation - The FFC system shall provided the user the capability to apply non-repudiation services with proof of origin (message signature) for electronic messages.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.4	
M-12	Non-Repudiation - The FFC system shall provide the user the capability to request non-repudiation services with proof of delivery (signed receipts) for electronic messages.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	8	5.1.4	
M-13	Confidentiality - The FFC system shall provide the user the capability to apply connectionless confidentiality for electronic messages	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	9	5.1.5	
M-14	Integrity - The FFC system shall provide connectionless integrity for electronic messaging	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	9	5.1.6	
M-15	Auditing - The FFC system shall provide the capability to trace and perform after the fact analysis of security-relevant actions performed by the FFC system components.	FFC System Description Rev FFC System Description Rev 2 May 6, 1996	9	5.1.7	

8.3 Theater Battle Management (TBM)

Table 16 identifies the applicable requirements extracted from the AF TBM Security Policy. Page and section numbers can be traced to the original document [30] and a reference field indicating any related documents.

Table 16. Theater Battle Management Requirements

Req. #	Description	Source	Pg.	Section	Reference
B-1	TBM must generate, exchange, and safeguard classified information from UNCLASSIFIED to TOP SECRET SCI, with clearances ranging from uncleared foreign nationals to TOP SECRET SCI.	TBM Security Policy	NA	3	
B-2	TBM must support dissemination and protection of nonreleasable US classified information.	TBM Security Policy	NA	3.6.3	
B-3	TBM systems that interface to or are used by our allies must be able to provide information that is releasable to those allies. The system may need to be releasable to those allies.	TBM Security Policy	NA	3.6.4	
B-4	Security guards, gateways, and firewalls may be used to mitigate risks of connections to affiliated systems.	TBM Security Policy	NA	3.6.5	
B-5	TBM security policy will need to follow the guidance of AFC4A Security Architecture.	TBM Security Policy	NA	4.2.2.2	?
B-6	TBM security policies and enforcement must be adaptable to planned contingencies.	TBM Security Policy	NA	4.5	
B-7	Each LSE should be capable of being accredited to provide a level of security commensurate with the classification of the information being processed.	TBM Security Policy	NA	5.2.1	
B-8	Each LSE should be capable of identifying and authenticating the identity of the individual user.	TBM Security Policy	NA	5.2.1.1	

Req. #	Description	Source	Pg.	Section	Reference
B-9	Access to information should limited to users with appropriate clearances and need-to-know.	TBM Security Policy	NA	5.2.1.2	
B-10	Each LSE should be capable of auditing security relevant events, may participate in distributed auditing services, and may collect network audit information.	TBM Security Policy	NA	5.2.1.3	
B-11	Each LSE should provide appropriate assurance that it meets its functional security requirements.	TBM Security Policy	NA	5.2.1.4	
B-12	Each LSE system's integrity should be periodically assured by hardware and software tools.	TBM Security Policy	NA	5.2.1.5	
B-13	Service availability is a primary functional requirement.	TBM Security Policy	NA	5.2.1.5	
B-14	Systems deployed in high-threat environments should offer overrun protection	TBM Security Policy	NA	5.2.2.1	
B-15	Each multilevel TBM system should be capable of creating and maintaining sensitivity labels for each mission data object, attaching each user's clearance to procession acting on their behalf, and enforcing access control based on sensitivity labels and clearances.	TBM Security Policy	NA	5.3.1	
B-16	A LSE should ensure that information output on single level channels is dominated by the sensitivity level of the channel.	TBM Security Policy	NA	5.3.2	
B-17	Each mission data object must have declassification conditions attached to it.	TBM Security Policy	NA	5.3.3	
B-18	Interfaces between systems of different levels must use procedures (e.g., air gaps) or products (e.g., guards) to provide sufficient level of assurance of adequate projection.	TBM Security Policy	NA	5.3.4	

Req. #	Description	Source	Pg.	Section	Reference
B-19	TBM systems that interface to, or are used by, our allies must be able to provide information that is releasable to those allies.	TBM Security Policy	NA	5.4	
B-20	A communications network supporting a LSE must provide protection of data against unauthorized modification. ¹	TBM Security Policy	NA	6.2.1	
B-21	A communications network supporting a LSE must provide protection of data against undetected loss, repetition, or insertion. ¹	TBM Security Policy	NA	6.2.1	
B-22	A communications network supporting a LSE must provide assurance of the identity of the sender, if required by the LSE security policy. ¹	TBM Security Policy	NA	6.2.1	
B-23	A communications network supporting a LSE must provide timely assurance to the sender that the data was received. ¹	TBM Security Policy	NA	6.2.1	
B-24	A communications network supporting a LSE must provide notification to the sending and receiving systems of any instance where assurance of the above policy goals was compromised due to loss of synchronization. ¹	TBM Security Policy	NA	6.2.1	
B-25	A communications network supporting a LSE must provide auditing capability such that all changes to network connectivity are audited. ¹	TBM Security Policy	NA	6.2.1	
B-26	A communications network supporting a LSE must provide assurance that the risk to the LSE is not increased. ¹	TBM Security Policy	NA	6.2.1	

Req. #	Description	Source	Pg.	Section	Reference
B-27	The TBM network security officer should monitor the TBM network to detect incidents that might indicate the propagation of a virus or worm.	TBM Security Policy	NA	6.3.1	
B-28	The TBM network security officer will maintain the set of classifications categories, and other markings that each receiving mission application should maintain.	TBM Security Policy	NA	6.3.1	
B-29	Each LSE should maintain audit records of security relevant events that occur as a result of interactions between an ES and any other ES with which the mission application system has a liaison.	TBM Security Policy	NA	6.3.2.1	
B-30	Each LSE should audit and report the addition or deletion of a physical attachment to the system to the appropriate Network Security Manager.	TBM Security Policy	NA	6.3.2.1	
B-31	Each LSE should audit and report changes in security parameters of or the intended use of any mission applications network interfaces to the appropriate Network Security Manager.	TBM Security Policy	NA	6.3.2.1	
B-32	Each LSE should audit and report any network-security error conditions associated with violations of secrecy and integrity requirements to the appropriate Network Security Manager.	TBM Security Policy	NA	6.3.2.1	
B-33	Each LSE should audit and report any events believed to imply a threat to the security posture of the TBM to the appropriate Network Security Manager.	TBM Security Policy	NA	6.3.2.1	

Req. #	Description	Source	Pg.	Section	Reference
B-34	Each LSE should audit and report the transmission of data in the clear due to loss of synchronization to the appropriate Network Security Manager.	TBM Security Policy	NA	6.3.2.1	
B-35	For each network security event, the audit record should include the identity of the involved systems, the date and time, and the type of event.	TBM Security Policy	NA	6.3.2.1	
B-36	Each system security officer is responsible for reviewing network and end user audit records and assessing the system's security posture.	TBM Security Policy	NA	6.3.2.2	
B-37	Each mission application system and internal interface should transmit the classification, categories, and markings identified by the NSM.	TBM Security Policy	NA	6.3.3	
B-38	Each TBM system must be prepared to protect against unauthorized access from other TBM systems or non-TBM systems. Firewalls may be used.	TBM Security Policy	NA	6.3.4	
B-39	Classified memory and storage media must be handled in accordance with DOD 5200.28-M.	TBM Security Policy	NA	6.3.5	DOD 5200.28-M
B-40	The TBM security architecture shall be consistent with the DoD goal security architecture.				
¹ The DGSA allocates only availability requirements to communications networks. For consistency with DGSA (Version 2.0), these security services should be allocated to the transfer system (which includes end system (ES) and relay system (RS) communication protocols, local communication systems (LSE) and the communications networks (CN)). This view is consistent with section 6.2.1's title: Communications Capabilities Between Mission Application Systems.					

8.4 Global Grid Security Architecture

Table 17 identifies the applicable requirements extracted from "Global Grid: The New Infosec Challenge" and discussions with NSA. Page and section numbers can be traced to the original document [31] and a reference field indicating any related documents.

Table 17. Global Grid Security Architecture Requirements

Req. #	Description	Source	Pg.	Section	Reference
G-1	High speed encryption for ATM and SONET	Global Grid: The New Infosec Challenge	1		
G-2	End-to-end security rather than "black box" (encryption) solutions	Global Grid: The New Infosec Challenge	2		
G-3	Secure, high speed multi-media services on a global scale	Global Grid: The New Infosec Challenge	1		
G-4	Security management/secure network management infrastructure	Global Grid: The New Infosec Challenge	2		
G-5	Base Global Grid on the goal security architecture (under consideration)	Global Grid: The New Infosec Challenge	2		
G-6	DISN is the operational recipient of global grid	Global Grid: The New Infosec Challenge	2		
G-7	Security management functions (such as key management) will permeate the global network and accommodate hierarchies of decentralization	Global Grid: The New Infosec Challenge	3		
G-8	Affordability and ease of use	Global Grid: The New Infosec Challenge	3		
G-9	Multiple levels of classification sensitivity	Global Grid: The New Infosec Challenge	4		
G-10	Allow appropriate access, prevent inappropriate access	Global Grid: The New Infosec Challenge	4		
G-11	Provide confidentiality, integrity, availability	Global Grid: The New Infosec Challenge	4		
G-12	Mobility "any user, any time, any place"	Global Grid: The New Infosec Challenge	4		

Req. #	Description	Source	Pg.	Section	Reference
G-13	Generation, transfer, storage and processing of multiple levels of sensitive and classified info	Global Grid: The New Infosec Challenge	4		
G-14	Authorized access of data should not be restricted by the sensitivity or classification of that information	Global Grid: The New Infosec Challenge	5		
G-15	Protection from unauthorized disclosure or modification	Global Grid: The New Infosec Challenge	5		
G-16	Global grid security architecture must consider interoperability, backward compatibility and existing infrastructure	Global Grid: The New Infosec Challenge	5		
G-17	Backward compatibility (algorithm selection, key negotiation and selection, placement in architecture)	Global Grid: The New Infosec Challenge	5		
G-18	Interoperability (algorithm selection, key negotiation and selection, placement in architecture)	Global Grid: The New Infosec Challenge	6		
G-19	Support for a variety of independent and integrated voice, data, video, facsimile, imagery applications (algorithm type and mode, key agility, multiple instantiations of crypto engines, programmable cryptography)	Global Grid: The New Infosec Challenge	6		
G-20	Support for multiple security levels (algorithms for various security levels, key agility, multiple instantiations of crypto engines, programmable cryptography)	Global Grid: The New Infosec Challenge	6		
G-21	Usability (automated key fill/benign fill, single point automated keying, automated remote control/status interfaces, downloadable software)	Global Grid: The New Infosec Challenge	6		
G-22	Reusability (modular cryptographic sub-elements)	Global Grid: The New Infosec Challenge	6		

Req. #	Description	Source	Pg.	Section	Reference
G-23	Certifiability (reusable certified cryptographic sub-elements, formal software development and CM practices, RED/BLACK separability)	Global Grid: The New Infosec Challenge	6		
G-24	Support for a variety of transmission technologies (algorithm type and mode, key agility, programmable cryptography, very high-speed cryptography)	Global Grid: The New Infosec Challenge	6		
G-25	Support for separation of multiple security levels within a security domain (high assurance software development and evaluation)	Global Grid: The New Infosec Challenge	6		
G-26	Separation versus integration with other system software (e.g., trusted appliques running on off-the-shelf operating systems, trusted operating system kernels with untrusted applications, trusted system/security management functions, trusted cryptographic control/bypass software)	Global Grid: The New Infosec Challenge	6		
G-27	Data encryption, digital signatures, authentication and data integrity	Global Grid: The New Infosec Challenge	7		
G-28	Security labels may need to be embedded in multiple protocol layers, e.g., IP & ATM. ATM will need the label to determine the appropriate key. Labels may also be required for security guards and upgrading/downgrading devices. Therefore, standards need to be developed to ensure security labels as available in consistent manner across all layers.	Global Grid: The New Infosec Challenge	8		
G-29	Security management of key material, authentication info, access control	Global Grid: The New Infosec Challenge	9		
G-30	Collection, analysis and reporting of security related events and audit info	Global Grid: The New Infosec Challenge	9		

Req. #	Description	Source	Pg.	Section	Reference
G-31	Detection of security compromise, conditions and coordination of recovery	Global Grid: The New Infosec Challenge	9		
G-32	Integrated network management across public and private nets	Global Grid: The New Infosec Challenge	9		
G-33	Denial of service detection, recovering	Global Grid: The New Infosec Challenge	9		
G-34	Common network management protocol with security	Global Grid: The New Infosec Challenge	9		
G-35	Priority routing	Global Grid: The New Infosec Challenge	9		
G-36	Secure multicast and broadcast to large numbers of subscribers	Global Grid: The New Infosec Challenge	10		
G-37	Develop protocols to get visibility up and down the protocol stack	NSA Discussion			
G-38	Different services offered at different layers	NSA Discussion			
G-39	Communication with other countries... processing for classification levels as well as handling labels; uniformity of labels; semantics, how do you know labels will be interpreted correctly	NSA Discussion			
G-40	Send messages between systems accredited at different levels, i.e., move back enclave boundary	NSA Discussion			
G-41	More sophisticated policy based mechanism same level = secure pipe, different policy at the other side = put in filter, changing between levels = put in a guard	NSA Discussion			
G-42	Assurance - trust that it is happening the way you want it to be from top to bottom	NSA Discussion			

8.5 Derived Requirements

The following table identifies the ATM USA requirements derived from the security requirements of the four Air Force designated programs specified above in this Appendix: the Global Grid Security Architecture, the Defense Information System Network Security Architecture (DISN), the Multilevel Information System Security Initiative (MISSI), and the Theater Battle Management C4I Architecture for Deployable Operations. In the traceability matrix given in Table 18 below, we make explicit the relationship between the ATM USA derived requirements and the requirements of these four Air Force designated programs. For each derived requirement, the related requirements from these programs are listed, where the programs are abbreviated as "G" (Global Grid), "I" (DISN), "M" (MISSI), and "B" (TBM) respectively. The referenced source requirements may be found in the other tables in this Appendix.

Table 18. Derived Requirements Traceability Matrix

Req. #	Source Requirements
ATM-1	I-2, I-3, I-4, I-5, I-6, I-8; M-3, M-4; B-7, B-11, B-26, B-39; G-2, G-22, G-23, G-24, G-36, G-40
ATM-2	I-1, I-2; B-20, B-21; M-13, M-14; G-11, G-15
ATM-3	M-1, M-6, M-7, M-8, M-11, M-12; B-22, B-23; G-12, G-27
ATM-4	I-1, I-2; M-5, M-9, M-10; B-8, B-38; G-10, G-15, G-33
ATM-5	M-15; B-10, B-24, B-25, B-29, B-30, B-31, B-32, B-33, B-34, B-35, B-36; G-30, G-31
ATM-6	I-1 ; B-13; G-11, G-33
ATM-7	B-1, B-9, B-15, B-16, B-17, B-28, B-37; G-9, G-13, G-20, G-25, G-28, G-39
ATM-8	B-4, B-18; M-2; G-41
ATM-9	G-8, G-35; R-2
ATM-10	G-35; R-1
ATM-11	G-4, G-7, G-29, G-32, G-34; I-12, I-13
ATM-12	B-5, B-40; G-5, G-16, G-17, G-19, G-21, G-26
ATM-13	B-2, B-3, B-19; G-18
ATM-14	G-1, G-3; R-3
ATM-15	I-9, I-10, I-11
NA	I-7 ; B-6, B-12, B-14, B-27; G-6, G-14, G-37, G-38

***MISSION
OF
AFRL/INFORMATION DIRECTORATE (IF)***

*The advancement and application of Information Systems Science
and Technology to meet Air Force unique requirements for
Information Dominance and its transition to aerospace systems to
meet Air Force needs.*